



A (kórházi) információrendszer modellje
doktori (PhD) értekezés tézisei

The model of the (hospital) information system
Ph.D. Thesis

Daragó László

Debreceni Egyetem
Informatikai Kar
Debrecen, 2006.

Tartalomjegyzék

1. Bevezetés.....	3
2. Irodalmi előzmények.....	5
3. A kórházi információrendszer modellje.....	9
4. Eredmények.....	22
5. Összefoglalás.....	26
The model of the (hospital) information system	
6. Premise.....	29
7. Literature antecedents	31
8. The model of the hospital information system.....	35
9. Results	47
10. Summary	51
11. Irodalomjegyzék / References	55
12. Publikációs lista / List of publications	59

1. Bevezetés

Jelen értekezés a kórházi információrendszerek kialakításának követelménytervezési fázisához ad módszertani útmutatót, az elméleti megoldást konkrét rendszeren bemutatva.

A kórházak információrendszereinek kialakulása Magyarországon is a gazdálkodás, elsősorban a főkönyvi könyvelés, anyaggazdálkodás, bérgazdálkodás területein alkalmazott, a gazdaság más szféráiból átvett alkalmazások implementálásával kezdődött a '80-as években. Az egészségügyi alkalmazások őse a regiszterprogramok voltak, amelyek alkalmazásával az aktuális állapotról legalább pillanatfelvétel volt készíthető, folyamatos működtetésük azonban nem sikerült, mivel a kezdeti lelkesedést az érdektelenség váltotta fel. Ezen a területen nem volt jelen a gazdálkodás területeire jellemző adó- és pénzügyi nyilvántartási és jelentési kötelezettség, így a betegellátással kapcsolatos alkalmazások kizárólag néhány orvos és a számítógépekkel bánni tudó műszaki vagy természettudományban jártas szakember együttműködésének köszönhetőek.

A személyi számítógépek és rendszereik, az egyszerűen telepíthető, olcsó adatbázis-kezelők elterjedésének köszönhetően minden orvosi területen megjelentek a számítógépes alkalmazások. Ezek - mivel a számítógép alkalmazásának bevezetése kórházsintű szabályozás híján az adott orvosi munkahely vezetőjének döntésén múlt - távolról sem nyújtottak egységes képet sem az alkalmazott szoftver platform, sem az alkalmazás funkciója, sem a kezelt dokumentumok tekintetében. Ugyanazon feladatok ellátására hasonló programok készültek, gyakran intézetben belül is különböző programot használtak az ugyanolyan profillal rendelkező különböző osztályok. A rendszerekkel szemben támasztott minőségi és megbízhatósági követelmények ugyanakkor alacsonyok voltak, hiszen elsődlegesen továbbra is a hagyományos, papír alapú dokumentációs rend volt érvényben, a számítógépes dokumentáció csak kiegészítő szerepet játszott.

A változást ezen a területen a teljesítmény elvű finanszírozási rendszer bevezetése hozta [5][47][34]. Ez a rendszer a gyógyítási tevékenységhez pénzügyi alapidokumentumot rendelt és adatszolgáltatási rendet írt elő a kórházak számára. A kormány adatszolgáltatás elrendelésekor annak teljesítéséhez PC-ket és ingyenes szoftvert is bocsátott a kórházak rendelkezésére. A mai napig van olyan kórház, amely ezen szoftveren túl semmilyen más egészségügyi alkalmazással nem rendelkezik. A kórházak egy része lépéseket tett integrált kórházi információrendszerek telepítésére, e rendszerek többnyire külföldön kifejlesztett rendszerek hazai adaptációi voltak. A jelenlegi helyzet szerint a gyógyító munkafolyamatokat, a gyógyítással kapcsolatos döntéstámogatást és a központi igazgatást egyaránt kiszolgálni képes rendszerek még mindig elvétve találhatóak meg a kórházakban. Sok intézet tett komoly

erőfeszítéseket akár máshol sikeresen működő rendszerek telepítésére, átütő eredmény nélkül. A rendszerek nem találkoznak a szervezet általános szimpátiájával, a velük szemben támasztott alapvető követelmények leegyszerűsödnek a teljesítmény elvű finanszírozási rendszerhez szükséges adatszolgáltatás biztosítására, és nem utolsó sorban a kórházi menedzsment érdekeltté tételére.

Az informatikai eszközök alkalmazása nem vont maga után az ipari termelésirányítási rendszereknél mérhető határfokjavulást az egészségügyi szakellátásban. Ennek mérése és bizonyítása egyszerűen elvégezhető: a szövődmények száma, az ápolási idő, de akár az intézeti eredmény szerinti elemzések sem mutatják eredményesebbnek az információrendszert használó intézeteket, mint az azt maximálisan mellőzőket. Ennek oka a rendszerek implementálásának folyamatában rendkívül fontos fázis elhagyása, nevezetesen a rendszerrel szemben támasztott követelmények szakszerű és reális igényeken alapuló megfogalmazása[11]. E helyzet kialakulása szükségszerű volt: az intézetek nem rendelkeznek a szükséges és aktuális szaktudással és/vagy szakértői kapacitással a követelmények meghatározásához, a rendszerek szállítóinak pedig ez nem érdeke. A rendszer szállítója a terméket minél alacsonyabb fajlagos költséggel akarja előállítani, magyarul, legfeljebb minimális változtatás mellett minél több helyen telepíteni. Ez viszont az eltérő konkrét igények és hagyományok (lásd később, például folklór) miatt kényelmetlenül használható, ezért aztán előbb-utóbb nem használt funkcionalitással bíró rendszerek telepítéséhez vezet.

A rendszertől elvárt követelmények meghatározása során alapvető fontosságú a tranzakciós rendszer adatbázisának meghatározásának alapjául szolgáló adatmodell meghatározása. A másik legfontosabb lépés azoknak a rendszertől elvárt funkcióknak a meghatározása, amelyek valóban kiszolgálják a gyakorlati munkafolyamatokat[10]. A gazdálkodás területén jól definiált - többnyire külső körülményeken alapuló - eljárási protokollok szerint zajlik a gyakorlat. A gyógyítás és ápolás munkafolyamatai és munkalistái nem egységesek, ezért csak a konkrét tevékenységek tanulmányozása után lehet meghatározni a rendszerrel szemben támasztott követelményeket. Azok az intézetek, amelyek rendelkeznek valamilyen minőségbiztosítási rendszerrel, ebben az esetben előnyt élveznek: nem feltétlenül azért, mert jó a szabályozás, hanem azért, mert a követelmények meghatározásához szükséges dokumentáció egy része rendelkezésre áll. A követelménydokumentáció összeállítása aztán változásokat kényszerít a minőségbiztosítási rendszeren is, a kórházak ettől ódzkodnak leginkább. Nehezen veszik tudomásul, hogy az információrendszer bevezetése beavatkozást jelent a napi munkafolyamatokba és a dokumentációs rendbe, így az azt leíró minőségbiztosítási rendszerbe is.

Jelen értekezés a kórházi információrendszer követelménydokumentációja meghatározásának folyamatát írja le a kórházi igények felmerülésétől a szoftverspecifikációhoz szükséges részletek meghatározásáig. Bevezeti a Felhasználó igény nevű dokumentációt, ami a Felhasználói követelmények nevű dokumentáció előzménye. Az adatmodell és a funkció-elemzés után a rendszer funkciói és a kezelt entitások közötti kapcsolatot a Martin-modellben[30][35], leírt c/u-mátrix¹ tulajdonságai határozzák meg. A dolgozatban a funkciók sorrendje a funkció-függőség (function dependency) modell és az életciklus-modell (life-cycle model) ötvözete szerint vannak meghatározva. E modell alkalmazása egyszerű munkafolyamatok esetén a P.C.I.² szervezési modell szerint egyszerű, azonban nagy bonyolultságú rendszerek esetén a konkurens folyamatokat nem tudja kezelni. Ezekben az esetekben gyakran vezet arra az ellenmondásos eredményre, hogy ugyanazon entitást több funkciónak is létre kell hoznia, ami a tranzakciós rendszerek esetén a rendszer lefagyások egyik leggyakrabban előforduló oka, ezért a rendszermodellekben tiltott. Mivel a funkció-függőség modell alkalmazása a gyógyító-ápoló tevékenységet kiszolgáló rendszereknél nem életszerű, a probléma megoldása az életciklus szerint modellben lévő, konkurens folyamatok körüli probléma megoldása. Az értekezés e probléma megoldása érdekében bevezeti a nem-szekvencia funkciókat, amelyek a rendszer részei ugyan, de nem részei szükségszerűen egyik életciklus szerinti folyamatnak sem. A c/u-mátrix kitöltésekor ugyanakkor ezen funkció-entitás kapcsolathoz rendelt „c”-vel és az ezen, nem szekvencia funkciókat beágyazó, itt bevezetett „h” jelöléssel megoldja a konkurens folyamatok körüli problémát.

Az értekezés a rendszerkövetelmény-dokumentáció elméleti felvázolása után az ott ismertetett szerkezet szerint valóságos kórházak által leírt igényekből kiindulva konkrét kórházi információrendszer rendszerkövetelmény dokumentációját tartalmazza, ahol a fenti, új megoldás alkalmazhatóságát és annak hatékonyságát is szemlélteti.

2. Irodalmi előzmények

A követelménytervezés módszere

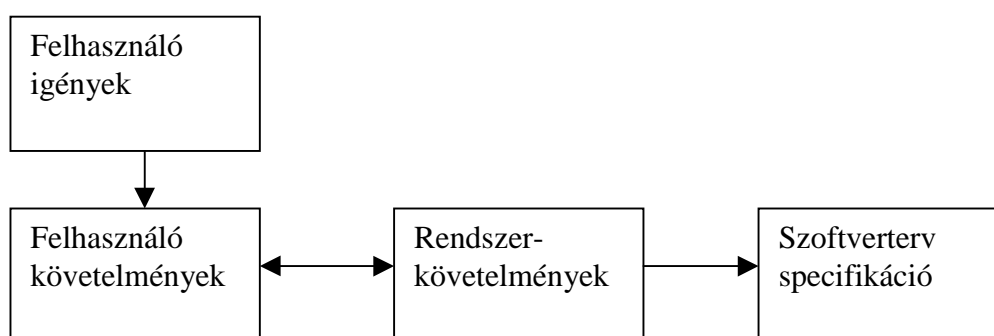
Egy információrendszerrel szemben támasztott elvárások általában rendkívül összetettek. A megoldandó problémák megértése bonyolult, a rendszertől elvárt működés nem egyértelmű a különböző felhasználók igényei, valamint egyéb, a rendszerrel szemben előírt elvárások szerint. A rendszer által nyújtandó szolgáltatások és a vele szemben támasztott megszorítások leírásait a *rendszer követelményei*, e szolgáltatások és megszorítások kitalálásának,

¹ Néhány irodalom CRUD mátrixnak nevezi

² Process Controll Information System

elemzésének, dokumentálásának és ellenőrzésének folyamatát a *követelmények tervezésének* nevezik [44].

Ha egy társaság (megrendelő) egy nagyméretű szoftverfejlesztési projektre szeretne szerződést kötni, eléggé absztrakt módon kell meghatározni az igényeit ahhoz, hogy a megoldás ne legyen előre adott. A követelményeket úgy kell meghatározni, hogy több szállító is árajánlatot tehessen a szerződésre, ezzel esetleg többféle módot kínálva az ügyfél szervezet igényeinek kielégítésére. Ha a szerződést az egyik szállító elnyerte, meg kell írnia egy részletesebb rendszerdefiníciót a megrendelő számára azért, hogy az megértse és jóváhagyhassa, amit a szoftver végezni fog. Mindkét dokumentumot nevezhetjük a rendszer követelmény dokumentumának



A követelmény dokumentumok fajtái:

- A *felhasználói igények* a megrendelő/felhasználó által, saját szavaival kifejtett elvárásait tartalmazza a telepítendő rendszerrel szemben. Általában nem teljes, emellett ellentmondásos, hiszen a felhasználó nem informatikai szakember, ugyanakkor az egyes munkatársai által megfogalmazott elvárások egyéni és sajátos érdeklődéseket vonnak maguk után. Ezek feloldása a *felhasználó követelmények* készítésekor megtörténik[10][44].
- A *felhasználói követelmények* ábrákkal, táblázatokkal, vázlatokkal és mintákkal kiegészített természetes nyelvű kijelentések arról, hogy mely szolgáltatásokat várunk el a rendszertől, és mely megszorítások mellett kell annak működnie. A *megrendelő* készíti.
- A *rendszerkövetelmények* a rendszerszolgáltatásokat és megszorításokat jelölik ki részletesen. A rendszerkövetelmény dokumentumnak, melyet néhol funkcionális specifikációnak is hívnak, javallott pontosnak lennie. Ez alapjául szolgálhat a rendszer vásárlója és a szoftverfejlesztő közötti szerződésnek. A *szállító* készíti a *megrendelő* által meghatározott követelmények és megszorítások figyelembe vételével.

- A szoftverterv-specifikáció a szoftverterv absztrakt leírása, a szoftver és hardver tervezéséhez és a megvalósításhoz szükséges dokumentáció, amely a részletesebb tervnek és az implementációnak az alapja. Ez a specifikáció további részleteket ad a rendszerkövetelmény-specifikációhoz. Tartalmazza mind a rendszer felhasználói követelményeit, mind a rendszerkövetelmények részletes specifikációját. A felhasználói és rendszerkövetelményeket egyetlen leírásba is össze lehet fogni, de a felhasználói követelmények a rendszerkövetelmény-specifikáció bevezetéseként megadhatóak. Ha a követelmények száma magas, a részletes rendszerkövetelmények külön dokumentumokként is elkészíthetők.

A szoftverkövetelmény-dokumentumnak hat követelményt ajánlott kielégítenie [23]:

- Csak külső rendszerviselkedést határozzon meg.
- Adjon meg a megvalósításra vonatkozó megszorításokat.
- Könnyen változtatható legyen.
- A rendszer karbantartói számára referencia eszközként szolgáljon.
- Tartalmazzon előrejelzéseket a rendszer életciklusára vonatkozóan.
- Jellemezze a nemkívánatos eseményekre adható elfogadható válaszokat.

Követelmények és megszorítások

A követelmények vonatkozhatnak a rendszertől elvárt konkrét szolgáltatásokra, végrehajtásuk módjára, valamint a rendszerrel, vagy egyes szolgáltatásaival kapcsolatos jellemzőkre, minőségi vagy teljesítménymutatókra. Mindkét kategóriában lehetnek olyanok, amelyek a működésnek a felhasználótól függetlenül létező szabályszerűségeit határozzák meg. A követelményeknek ezt a felosztása az alábbiak szerint történik[44]:

Funkcionális követelmények

A rendszer által biztosított szolgáltatásokat, funkciókat írják le. Tartalmazzák a rendszernek adott szituációban való viselkedését, a rendszer algoritmusait, néhány esetben azt, hogyan nem szabad viselkednie a rendszernek meghatározott esetekben.

Elvben a rendszer funkcionális követelmény specifikációjának mind teljesnek, mind ellentmondásmentesnek kellene lennie. A teljesség azt jelenti, hogy a felhasználó által igényelt összes szolgáltatás definiálva legyen. Az ellentmondás-mentesség azt jelenti, hogy a követelményeknél ne legyenek ellentmondó meghatározások. A gyakorlatban, nagyméretű, összetett rendszereknél nagyon nehéz a követelmények teljességét és ellentmondás mentességét elérni. Ennek oka részben a rendszer belső összetettsége, részben pedig az, hogy

a különböző nézőpontok egymásnak ellentmondó igényeket jelentenek. Elképzelhető, hogy ezek az ellentmondások nem látszanak azonnal a követelmények első megadásakor[18][52].

A funkcionális követelmény specifikáció teljességének és ellentmondás mentességének megvalósítására alkalmazható a *funkció-entitás kapcsolatrendszer ábrázolásának* módszere (c/u-mátrix). A funkcionális elemzés során a rendszertől elvárt szolgáltatások meghatározása történik. Az ilyen módon meghatározott funkciókhoz a rendszer által használandó entitások lesznek hozzárendelve.

Nem-funkcionális követelmények

A rendszer által kínált funkciókra és szolgáltatásokra, vagy akár magára a rendszerre tett, megszorítások. Magukban foglalnak időbeli korlátozásokat, a fejlesztési folyamatra tett megszorításokat, szabványokat, stb. Nem közvetlenül a rendszer által szállított, a működéstől elvárt szolgáltatásokkal, mint specifikus funkciókkal foglalkoznak. Vonatkozhatnak olyan rendszertulajdonságokra, mint a megbízhatóság, a válaszidő és a tárfoglalás, a konfiguráció, az egyes rendszerkomponensekkel szemben támasztott követelmények (pl. I/O kapacitás), vagy az adatinterfészek leírásai.

Számos nem-funkcionális követelmény a rendszer egészére vonatkozik, nem pedig egyedi rendszerösszetevőkre. Ezek általában fontosabbak, mint az egyedi funkcionális követelmények, hiszen amíg egy egyedi funkcionális követelmény teljesítésének sikertelensége gyengítheti a rendszert, addig egy nem-funkcionális rendszerkövetelmény teljesítésének kudarca a teljes rendszert használhatatlanná teheti [43], [28].

Egy kórházi információrendszer, amely nem a megfelelő mértékegységet használ a diagnosztikai vizsgálatok eredményeinek tárolására és közlésére, az orvost számára haszontalan adatokkal halmozza el a szükséges információ helyett. Ebben az esetben a diagnosztikai gyógyító munka támogatására nyilvánvalóan alkalmatlan.

Hasonlóan, amennyiben a teljesítmény-finanszírozási rendszer előírt adatszolgáltatási kötelezettségét nem teljesíti a fogadó által előírt adatformátumban, nem képes a legalapvetőbb követelmény, nevezetesen a kórház bevételének dokumentálására.

Amennyiben a járóbeteg-szakellátás információs rendszere a beteget távozásakor nem tudja ellátni az ellátási esetről készült dokumentációval, az egészségügyi dokumentáció szabályainak nem képes megfelelni, vagyis használhatatlan.

A nem-funkcionális követelmények nem feltétlenül a konkrét szoftverrendszerre vonatkoznak. Meghatározhatja például, hogy a követelménydokumentációt és a szoftver specifikációt a Martin-féle szervezési modell nyelvezete és szerkezete szerint, az UML

dokumentációs szabvány alkalmazásával kell készíteni és karbantartani [4][40]. Hivatkozhat az alkalmazott minőségbiztosítási rendszer előírásaira, illetve belső szabályzókra, vagy külső jogszabályokra, egyéb rendszerekhez való igazodásra. Lehetnek speciális felhasználói igények, alkalmazandó algoritmusok, adatkezelési rendelkezések[4].

Szakterületi követelmények

A rendszer alkalmazási szakterületéről származó, a szakterület jellegzetességeit tükrözik. A szakterületi követelmények egyaránt lehetnek funkcionális vagy nem-funkcionális követelmények. Nem a felhasználók egyedi igényeiből, hanem a szakterületen alkalmazandó szabályszerűségeket, algoritmusokat, megszorításokat tartalmazza. Ezek a követelmények az alkalmazási terület alapvető megfelelőségi feltételeit tartalmazzák, így teljesülésüktől a kielégítő működtethetőség függ.

Művese kezelés rendszeréről szóló szakterületi követelmény

A művese kezelés minőségi mutatói közé tartozik a Kt/V érték. Értéke szoros korrelációban áll a beteg várható élettartamával.

$$Kt/V = -\ln\left(\frac{CN_2}{CN_1} - 0,03\right) + \left(4 - 3,5 * \frac{CN_2}{CN_1}\right) * \left(\frac{UF}{BW}\right),$$

ahol CN_1 a vér dialízis előtti karbamid-nitrogén tartalma

CN_2 a vér dialízis utáni karbamid-nitrogén tartalma

UF az adott dialízis ultrafiltrációja, literben

BW a testsúly a dialízis után, kg-ban

A szakterületi követelmények az alkalmazási szakterületre jellemző nyelven vannak kifejezve, és a szoftvertervezők számára gyakran nehéz megérteni őket. A szakterület szakértői egyszerűen elhagyhatnak információt a követelményből, mivel az számukra teljesen nyilvánvaló. Ugyanakkor a rendszer fejlesztői számára az lehet, hogy nem nyilvánvaló, és előfordulhat, hogy emiatt a követelményt nem kielégítő módon valósítják meg. Éppen ezeknek a félreértéseknek az elkerülése érdekében fontos az *adott szakterületen és annak információs rendszereiben* jártas ügyvitelszervező szakemberek (computing professionals) bevonás a tervezés folyamatába[11].

3. A kórházi információrendszer modellje

A dolgozatban bemutatott modell magyar kórházak által megfogalmazott felhasználói igényekből és követelményekből indul ki. Ezek mellőzik a konkrét kórházi környezetre

vonatkozó utalásokat, ugyanakkor a tipikus magyar körülményekre jellemző felhasználó igényeket fogalmazzák meg. A rendszertől elvárt tulajdonságok felsorolása az orvosi és gazdasági, valamint gyakorló kórházi információtechnológiai szakemberek közreműködésével történt[14][51]. A dolgozat további része a (minta-) rendszer követelménydokumentációját, a rendszerkövetelmények meghatározásának lépéseit tartalmazva írja le. Az egyes szakaszok bevezetéssel és a módszer ismertetésével, majd a konkrét adatok, eredmények leírása után értékeléssel, amennyiben szükséges, magyarázatokkal végződnek.

A kórháznak a rendszerrel szemben támasztott elvárásai

A rendszer tervezésének első szakasza az igény megfogalmazása. A különböző szakterületeken felmerülő információtechnológiai igények összeállítása természetes nyelven megfogalmazott igényekből áll. Nem szabad elfelejteni, hogy a kórház nem szoftverfejlesztő társaság, így nem várható el a szabatos, információtechnológiai szakemberek számára egyértelmű megfogalmazás.

A természetes nyelven írt követelményeknél ugyanakkor problémák merülhetnek föl, ezért ezek elkerülésére ajánlott odafigyelni:

- *Az egyértelműség hiánya*

Olykor nehéz a nyelvet pontos, egyértelmű módon használni anélkül, hogy a dokumentumot terjengőssé és nehezen olvashatóvá ne tennék.

- *Követelmények keveredése*

A funkcionális követelmények, nem-funkcionális követelmények, rendszercélok és tervezési információk nem különíthetők el tisztán.

- *Követelmények ötvöződése*

Több különböző követelmény egyetlen követelményként lehet megadva.

Az igényelt szolgáltatások

A felhasználói követelmények a funkcionális és nem-funkcionális követelményeket írják le úgy, hogy azok részletes technikai ismeretek nélkül is érthetőek legyenek. A rendszernek a felhasználótól elvárt működését írják le a belső működés mellőzésével. A felhasználói követelményeket nem javallott implementációs modell segítségével definiálni, hanem természetes nyelven, úrlapok és egyszerű, könnyen értelmezhető diagramok segítségével kell őket megírni. A felhasználói követelmények tervezésénél érdemes bevonni olyan szakembereket, vagy szakértő céget, amely - a kórházi szakemberek információtechnológiai vagy informatikai felkészültségét vagy teljesítőképeségét általában meghaladó - tevékenység során naprakész rendelkezik[11]. Az ilyen tanácsadók a tenderkiírás

fázisában általában hasznosnak bizonyulnak, bár sok esetben felmerül a gyanú, hogy valamely szállító előretolt felderítői.

Sok megrendelő esik abba hibába, hogy túlságosan mélyen és részletekbe menően akarja meghatározni a rendszer belső működését. Ezzel nemcsak a szállító válláról veszi le a felelősség terhét nem kielégítő működés esetén, de saját, - a szoftver- és hardverfejlesztőétől nyilvánvalóan alacsonyabb szintű - ismeretei a fejlődés gátját is szabják. Az is elképzelhető, hogy a hogyan meghatározása olyan ellentmondást eredményez a rendszerkövetelményekben, ami a nem-funkcionális követelmények teljesíthetetlenségét vonja maga után. Ezek alól természetesen kivételek lehetnek a szakterületi követelmények, a külső követelmények, vagy a rendszer dokumentációjára, nyelvezetére vonatkozó előírások, amennyiben lehetetlen eltekinteni a tervezés folyamatába történő beavatkozástól.

A felhasználó igények elemzése és egyértelműbbé tétele vezet a felhasználó számára igényelt szolgáltatások meghatározásához[10].

A rendszerkövetelmények

A rendszerkövetelmények a felhasználói követelményeken alapuló olyan részletes leírások, amelyek a felhasználó és a szállító közötti, a rendszerről történő kommunikáció alapja. Ez a megvalósítási szerződés tárgya, ezért az egész rendszert összefüggéseiben kell, hogy tartalmazza. A szoftver-specifikáció dokumentum alapja, a szoftvertervezők számára a rendszerterv kiindulási pontja. Tartalmazhatja a rendszer különböző modelljeit, pl. egy objektummodellt vagy egy adatfolyam-modellt. A rendszerkövetelményeknek azt kell leírniuk, hogy *mit* kell csinálnia a rendszernek, nem pedig azt, *hogyan* kell ezt megvalósítania.

A rendszerkövetelmények meghatározásának eszközei

A felhasználó követelmények - nevéből is következően - a felhasználó (megrendelő) által meghatározott elvárásokat tartalmazza. Koránt sem teljes és ellentmondásmentes, ami elsősorban a felhasználó szubjektív értékítéletének és rendszer ismereteinek következménye. A rendszerkövetelményeknek ugyanakkor teljesnek és ellentmondás-mentesnek kell lennie, hiszen ez egyrészt a rendszer megvalósítási szerződésének alapja, másrészt a szoftverterv specifikációé. A szállító (fejlesztő) ezért részletekbe menően is meg kell, hogy vizsgálja a rendszertől elvárt külső működést, és az implementálhatóság feltételeit.

A kiszorgálandó szervezet feladatrendszerének leírása

A szállítandó rendszernek a megrendelő (rész-) tevékenységét kell kiszorgálnia. A kiszorgálandó tevékenység nem vonatkozatható el az azt végző szervezettől, hiszen a rendszert alkalmazó szervezet egyes konkrét tevékenységeit maga a rendszer milyensége nagymértékben meghatározza, ugyanakkor a szállítónak nem szabad elfelejtenie, hogy a

rendszer egy jól meghatározott feladatrendszerrel bíró szervezet kiszolgálására hivatott. A feladatrendszert az információrendszer tervezése előtt át kell tekinteni, aktualizálni kell [45]:

1. Küldetés (Mission) - a tulajdonos határozza meg. Tömör, magas szintű megfogalmazása a (rendszert alkalmazó) szervezet szándékának és politikájának. (Ki vagyok és miért vagyok?)
2. Stratégia (Strategy) - Politikák és tervek vázlata, amely meghatározza a szervezet működését adott időszakban (nincs jól meghatározott eleje és vége). Meghatározza a szervezet környezete és a küldetése. A stratégiát mindig a vezetés határozza meg. (Hogyan csinálom? Milyen piacot célok meg? Milyen erőforrásokat használok?)

A környezet alatt a működést befolyásoló minden tényezőt fel kell sorolni, hiszen ezek megléte, mibenléte és változásai alapvetően és érdemben befolyásolják a rendszert. Ilyen tényezők:

- Politikai környezet (Kiszámíthatóság, jogbiztonság, korrupció, stabilitás).
 - Infrastrukturális környezet (telekommunikáció, logisztika, technikai feltételek).
 - Jogi környezet (törvények, egyéb (pl. helyi) jogszabályok)
 - Erőforrások (műszaki, pénzügyi, materiális, humán).
3. Kívánalmak (Objectives, Vision) - Általános megfogalmazása azon irányoknak, amelyek az egyes részterületeken folyó munkát meghatározzák konkrét időpont meghatározása nélkül. Jövőkép, amihez a stratégia vezet. (Milyen irányban kívánok haladni?)
 4. Célok (Goals) - Konkrét célok meghatározása, amelyek mérhetőek, konkrét határidővel rendelkeznek. Ezeken keresztül mérhető le az is, hogy a kívánalmak teljesülése felé halad-e a szervezet.

A célok közül érdemes megkülönböztetni az állandó, vagy folyamatos, a taktikai, vagy rövidtávú és a stratégiai, vagy hosszú távú célokat. Az is előfordul, hogy a taktikai célok egy része a stratégiai célok részletezéseként értelmezhető, a stratégiai célok pedig magának a küldetésnek a pontokba szedett megfogalmazásai. Az állandó célok a folyamatosan, újra és újra teljesítendő célokat jelentik. Ilyen a kórház esetén a betegellátás hatásossága, valamint hatékonysága.

5. Kritikus sikertényezők (Critical success factors: csf's) - Olyan kulcsfontosságú tényezők, amelyek szükségesek ahhoz, a célokat eléréséhez, a kívánalmak és a küldetés teljesüléséhez. Ezek mérhető faktorok, indikátorok. A kritikus sikertényezőket is szokás súlyosság szerint osztályozni, azaz külön kezelni és megkülönböztetni a „kritikus”, „fontos”, „előnyös” sikertényezőket. Ez utóbbi osztályozás a sikertényező teljesülése és a teljesüléshez szükséges ráfordítások közötti eltérések érzékeltetésére jó módszer[22].

6. Mérések - A célok, sikertényezők ellenőrzése, szisztematikus és rendszeres mérések végzése. (pl. cél elérése adott időpontban) Szisztematikus: a mérést ugyanúgy kell minden alkalommal végezni. Rendszeres: nem spontán, hanem tervezett időpontban történik.
7. Problémák - Problémák a mérések vagy a célok el nem érése eredményeképpen adódhatnak. Fel kell készülni ezekre a problémákra, tudni kell kezelni. A probléma nem jelent feltétlenül „rossz” dolgot. Lehetnek külső és belső problémák. A külső problémák a szervezeten kívüli körülmények alakulásából adódnak; ilyen lehet a küldetés változása, vagy a környezet változásai. A külső problémák kezelése legtöbbször a stratégia, a célok változásai vonják maguk után (profil, költségvetés módosítása). A belső problémák okai a szervezeten belül keresendők, a rendszertől elvárható, hogy mind a belső, mind a külső problémák esetén aktív szerepet játsszon a döntés előkészítésben.

A problémák kérdéskörébe közé tartoznak a rendszer rendelkezésre állásával kapcsolatos előkészületek is, a tervezett leállások előkészítése és a nem tervezett leállásokra, helyzetekre történő felkészülés, a katasztrófatervek készítése.

A rendszerkövetelményeknek tartalmazniuk kell a méréseket megvalósító funkciókat, mivel azok a felhasználó átfogó célkitűzéseinek megvalósításához szükségesek. A felhasználói igények ezekre a mérésekre szoktak utalni, de konkrét megfogalmazásuk hiányában a felhasználói követelmények közül sokszor kimaradnak[12].

Megjegyzés

Minden mérés egy, vagy több entitás előállítását jelenti. A rendszertől elvárt szolgáltatások listáját a mérésekkel bővíteni kell.

A követelménytervezési folyamatnak az előzőek szerint a megvalósíthatósági tanulmánnyal kellene kezdődnie. E tanulmánynak tartalmazni kell a megrendelő, jelen esetben a kórház feladatrendszerét. Azáltal, hogy a feladatrendszerből kiindulva történik a szállított információs rendszer, a rendszer más szervezetben történő használhatósága könnyen megvizsgálható. Amennyiben az új szervezet feladatrendszere eltér a konkrét szervezetétől, a meglévő és a szükséges rendszerek közötti különbség is meghatározható.

Ugyancsak e fázis célja eljutni a működéshez elengedhetetlen kritikus sikertényezők, illetve az azok méréséhez szükséges funkciók és entitások meghatározásához. A mérések az információs rendszer jelentései lesznek. Elképzelhető, hogy a megrendelő az egyes sikertényezők mérését nem kéri, sőt az is előfordulhat, hogy a megrendelő egyetlen ilyen jelentés megvalósítását sem kéri szállítani. A szállítónak ennek ellenére meg kell ezeket határozni, hiszen a rendszerkövetelmény nemcsak a megvalósított, hanem a kizárt funkciókat is tartalmazhatja, azaz az, hogy a rendszer milyen funkciót támogat, és miket nem, a dokumentáció teljességét és egyértelműségét javítja.

Funkcionális dekompozíció (FD)

A funkcionális követelmények meghatározása során felsorolt funkciók elsődleges és másodlagos főfunkciókra (funkció csoportokra) oszthatóak. Az elsődleges főfunkciók közé a szervezet tevékenységét meghatározó, annak profiljára jellemző funkciók tartoznak, míg a másodlagos főfunkciók közé a többi, az elsődleges főfunkciók közé nem tartozó funkciók. (Ez a felosztás a meglévő rendszerek újrafelhasználása, illetve a más rendszerekkel való kommunikáció szempontjából hasznos: a másodlagos főfunkciók megvalósításához teljesen más szervezet rendszereit is felhasználhatók, míg az elsődlegeseknél csak az ugyanilyen profillal rendelkező szervezetek rendszerei.)

A funkcionális dekompozíció a szervezet tevékenységét az alábbiak szerint részletezi [29]:

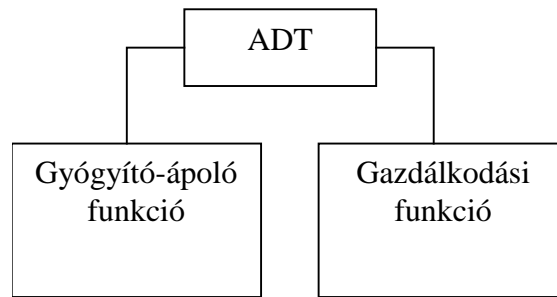
- Főfunkció (vagy funkciócsoport): valamely tevékenységi területtel kapcsolatos szolgáltatások.
- Funkció: a főfunkción belüli konkrét szolgáltatás. A funkció és főfunkció valamely folyamatos, meglévő szolgáltatást ír le, de nem konkrét tevékenységet. A funkció esetleg további funkciókra osztható.
- Folyamat: A funkció részeként leírható konkrét tevékenység. A folyamat eljárásokból álló, inputjában, outputjában, jól meghatározott tevékenység, jellemző rá, hogy - a funkciókkal ellentétben - van kezdete és befejezése.

(A továbbiakban mindhárom szolgáltatás funkcióként szerepel ott, ahol nem szükséges megkülönböztetésük.)

Ez a felsorolás azt részletezi, milyen szolgáltatást nyújt az adott funkció, de nem tér ki arra, hogyan teszi azt. Ez utóbbit a szoftverterv-specifikáció tartalmazza majd, amely a folyamatokat alkotó eljárásokat írja le[20][41][53].

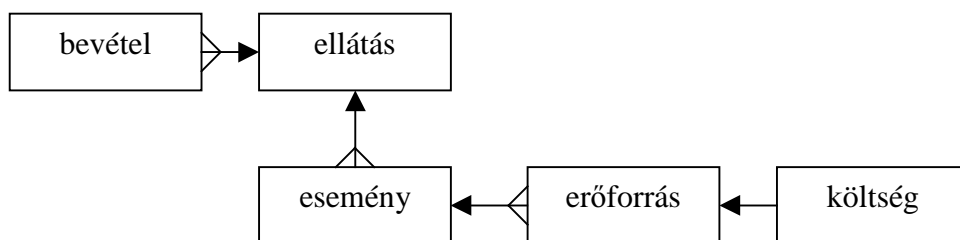
A felhasználó követelményekben felsorolt funkciókat a funkcionális dekompozíció szerint elrendezésben ábrázolva nyilvánvalóvá válnak az esetleges párhuzamosságok, illetve előtűnnek a nem megfelelő részletességgel meghatározott funkciók, vagyis azok, amelyek nincsenek folyamatokra bontva. A folyamatra bontás lényege az, hogy egy szolgáltatás igénylése csak annak konkrét és részletes meghatározásával, azaz folyamatokra történő bontásával történhet. A rendszerkövetelmények tervezése során a hiányzó részletezést el kell végezni, ellenkező esetben az igényelt szolgáltatást a rendszer funkciói közül törölni kell.

A kórházi rendszer architektúra modellje a gyógyító-ápoló tevékenységet végző gazdálkodó szervezet modellje:



A betegellátás adatszolgáltatási rendszere a konkrét ellátási esetek dokumentálását, az ellátási folyamat során felmerülő döntéstámogatást szolgálja ki. A gyógyító-ápoló tevékenység során felhasznált erőforrásokot az ellátási esethez kell rendelni az orvosi, ápolási dokumentáció szakmai szabályai szerint. Ezzel párhuzamosan a gazdálkodási szempontból az egyes ellátási esetek a finanszírozási rendszer szerinti dokumentálása pénzügyi alapbizonylatnak tekintendő, azaz az ellátási eset a finanszírozási rendszer szabályai szerint bevételi összeggel azonosítható. Ezzel a bevétellel állítandó szembe az ellátás során felhasznált erőforrások költsége, amelyek a gazdálkodás szabályai szerint közvetlen, vagy közvetett költségnek tekintendők és elszámolandóak.

A betegellátás leegyszerűsített adatmodellje:



A fenti ábra kapcsolatai például így olvashatók ki:

- „Egy ellátási esethez 0, vagy több esemény tartozik”.
- „Egy erőforráshoz 0 vagy egy költség tartozik”.

A jelölés részletes magyarázata a 3.2.2.4. Az adatmodell fejezetben.

Az ellátási eset adatkezelésére szolgáló ADT-rendszer, azaz a betegfelvételi-, elbocsátási és áthelyezési rendszerhez (alrendszerhez) kapcsolódnak az ellátással összefüggő események, azaz az orvosi-, ápolói dokumentáció, másrészt az esetet rendelhető az ellátásért kapott térítés, és az ellátás során közvetlenül, vagy közvetve felhasznált erőforrásokhoz rendelt költség.

A rendszer architektúrája tehát a betegellátás orvosi adatkezelését kiszolgáló modulra, a kórház gazdálkodását támogató modulra és ezeket a modulokat integráló alrendszerre tagolható.

Megjegyzés

A felhasználói követelmények ismeretében a szállító feladata valamely alapmodell (koncepció) szerint a részletesebb, rendszerkövetelmények meghatározása. Le kell írni a megrendelő szervezetének hierarchia szerinti felépítését, és fel kell vázolni a cég feladatrendszerét.

A megrendelő ezeket a feladatokat feleslegesnek tarthatja, sőt nehezményezheti. A fejlesztőnek (szállítónak) ugyanakkor nem szabad elfelejtenie, hogy a rendszert alkalmazó megrendelő sikertelensége egyben az ő sikertelensége is. Minden esetben, amikor a megrendelő kitér a fejlesztés egy-egy fázisa előtt, a szállító érdeke a fázis elmaradása körülményeinek korrekt dokumentálása.

Feltűnő, hogy míg a gazdálkodással kapcsolatos szolgáltatások a jelenlegi szervezeti felépítésnek megfelelően csoportosítva részletesen szerepelnek, addig a betegellátással kapcsolatos szolgáltatások csak elnagyolva. Ez a másodlagos funkcióknak az előtérbe helyezését sejteti, azaz a Parkinsoni túlbürokratizálódás tüneteit mutatja.

Ennek történeti oka, hogy az információtechnológia alkalmazása a magyar (és külföldi) egészségügyi ellátórendszerben a gazdálkodási, nevezetesen a pénzügyi, számviteli rendszerek beszámolási kötelezettségeinek elrendelésével kezdődött. Ez a kívülről jövő hatás a kórházak számára kötelezően alkalmazandóvá tette az elektronikus adatkezelést annak gazdaságosságától függetlenül. A betegellátással kapcsolatban ilyen kényszer nem merült fel, ezért ez a terület kiforrotlanabb. Jelenleg sincsenek kialakítva Magyarországon az egészségügyi ellátás minőségi kritériumai, így az ennek történő megfelelési kényszer sem hat az információtechnológia alkalmazása irányába. Két olyan terület van, ahol alkalmazási kényszer lépett fel: a teljesítmény adatszolgáltatás végzése, amely a működést meghatározó bevétel túlnyomó részének előállításához szükséges (s amely bevezetése a mai napig az ellátó személyzet heves ellenállásába ütközik), a másik pedig az elektronikus kórlapvezetés, amely az egészségügyi személyzet adminisztrációs terheit csökkenti.

A kezelt entitások meghatározása

A rendszerkövetelmények meghatározásának lényeges eleme a rendszer által kezelt entitások felsorolása.

Érdekes az egyes, felsorolt entitásoknál feltüntetni, honnan származnak (külső entitások, vagy valamely funkció állítja elő) és azt is, mire használják őket (külső entitások,

vagy valamely funkció működéséhez szükséges). A fordított módszer is hasznos, azaz az egyes funkciók esetén felsorolni azokat az entitásokat, amelyeket felhasználnak, vagy létrehoz.

A külső entitások igazából nem részei az adatmodellnek - már felhasználói követelmények leírásakor fel kell sorolni őket - de ellenőrzésre kiválóan felhasználhatóak: a külső entitások minden attribútuma kell, hogy szerepeljen valamely belső entitás attribútumaként az adatmodellben. A külső entitások a rendszeren kívüli kommunikációt megvalósító adatok: lehetnek bejövő, vagy kimenő entitások. Egy részük a megrendelő szervezet és a külvilág párbeszédjének leírására szolgál, míg vannak olyan entitások, amelyek a rendszeren kívüliek (külső-külső: kk), de a szervezeten belüli adatcserét szolgálják (külső-belső: kb). Az előbbiek közé tartoznak a különféle beszállítókkal, vásárlókkal, hatóságokkal folytatott adatcsere, míg az utóbbiak a szervezet egyéb alrendszereivel történő adatcserét, vagy például az archiválási funkciókat valósítják meg. A két csoportot érdemes megkülönböztetni.

A külső-belső entitások felülvizsgálatát rendszeresen javasolt elvégezni. A kifelé menő külső-belső entitásoknál meg kell jelölni, hogy az archívumba kerül, vagy a szervezeten belül jól meghatározott felhasználóhoz. Ez utóbbinak a szervezet organogramján szerepelnie kell. Az archiválendő adatokat a karbantartott archiválási rendszer írja le. A felhasználóhoz kerülő külső-belső entitások sorsát, azok szükségességét rendszeresen felül kell vizsgálni. Gyakori, hogy új formai vagy tartalmi követelményt írnak elő az ilyen entitásokra, miközben a korábban használtat nem helyezik hatályon kívül.

Az egészségügy járóbeteg-szakellátás finanszírozási rendszerének alapidokumentuma az Ambuláns adatlap. A bevezetés után 6 évvel még mindig volt olyan kórház, ami az Ambuláns adatlap bevezetése előtt használt Ambuláns naplót is használta, mivel még volt felhalmozott készlet a raktárban. A raktár, mivel rendszeresen használták a nyomtatványt, időről időre utánnymást rendelt. A kórházban senkit nem zavart, hogy a kitöltött Ambuláns naplót sehová nem kellett eljuttatni, csak az Ambuláns adatlapról rendelkezett az egészségbiztosító.

A külső-külső entitások valamely jogszabály, vagy egyezmény keretei között meghatározott formai és tartalmi megszorításokat tartalmaznak, így ezek változtatására nincs lehetőség. A külső-belső entitások a megrendelő szervezetén belül vannak meghatározva az üzleti-, minőségbiztosítási-, vagy termelési szabályzatokkal, így a rendszer által nyújtott többlet szolgáltatások esetén az entitások megváltoztatása a megrendelő szervezetén belüli döntés kérdése. Lényeges feljegyezni, hogy mely funkciók állítják elő a kifelé menő külső entitásokat, illetve mely funkciók használják fel a bejövőket[2]. Az *External diagram* jó eszköz a grafikus ábrázolásra:

A belső entitások meghatározása felhasználó által a rendszer bevezetése előtt használt entitások felsorolásával kezdődik. További entitásokat határoznak meg a feladatrendszerből következő mérések végzéséhez szükséges entitások, és a funkcionális követelményeknél felsorolt a funkciók inputjainak és outputjainak felsorolása.

Érdeemes megjegyezni, hogy a rendszer függvényei és eljárásai is funkciók. Ebben a táblázatban minden entitás, így a külső entitások is szerepelnek. A külső és belső entitások nem diszjunkt halmazok, azaz egy entitás nem külső *vagy* belső entitás, hanem lehet külső *és* belső entitás is.

A funkciók és entitások kapcsolata, a c/u mátrix

Egy szervezet számára az új információrendszer bevezetése szükségszerűen a meglévő munkarend, a hagyományos feladatmegosztás átalakítását vonja maga után. A rendszer bevezetése funkció átcsoportosításokkal és átalakításokkal, azaz új feladatok, funkciók megjelenésével jár, ugyanakkor más funkciók megszűnését vonja maga után. Amennyiben ez nem így van, kérdésessé válik a hatékonyabb működés elérése. A c/u mátrix műveletei a szervezet funkcióinak és entitásainak átalakítását áttekintését és rendszer szempontjából optimális átcsoportosítását, átalakítását végzik.

A c/u mátrix funkciók és a rendszer entitásai közötti kapcsolatot ábrázolja, a tervezési modell legfontosabb eleme. A mátrix soraiban a funkciók szerepelnek, míg az oszlopokban az entitások [36][30].

A mátrix sorainak kitöltésekor feltételezhető, hogy a funkciók folyamat mélységig részletezve lettek. Nem nagy hiba, ha ez mégsem történt meg, mert a későbbi művelet a nem kielégítő módon részletezett funkciókat kimutatják majd. Kétféle folyamat különböztethető meg:

- Speciális, vagy szekvencia folyamat: a funkción belül jellemző rá a többi folyamathoz képest az időbeni pozíciója. Ilyen egy raktári anyag kiadása, amely csak az anyag utalványozása előtt történhet, vagy a beteg diagnózisának felállítása a terápia elrendelése előtt. Az időbeni sorrend meghatározása az u.n. life-cycle elv szerint, vagyis a tapasztalati élet-, vagy folyamatciklus alapján történik.
- Általános, vagy beágyazott folyamat: több funkcióban is feltűnhetne, egyértelműen nem helyezhető el az életciklus szerinti sorrendben. Ezeket a folyamatokat a speciális folyamatok részeként, azokba beágyazva lesznek használva. Ilyen például a beteg számára végzett beavatkozás végzése, amely az egymástól jól elkülönült járóbeteg- és fekvőbeteg ellátás során is történik. Az általános funkciókat A/...-val kezdődő nevük különbözteti meg.

A c/u mátrix műveleti közül a legelső a mátrix kitöltése, amely a funkció-entitás segédtábla alapján történik.

A funkciók és entitások lehetséges kapcsolatai:

- a funkció csak felhasználja az entitást, de műveletet nem végez rajta: r (read)
- a funkció létrehozza az entitást: c (create)
- a funkció felhasználja és módosítja az entitást: u (use)
- a funkció törli az entitást: d (delete)
- a funkció közvetett módon hozza létre az entitást, azaz tartalmazza az entitást létrehozó általános funkciót: h (have create)

A c/u mátrix műveletei

11. A c/u mátrixot a felhasználói, illetve megléte esetén a rendszerkövetelmény dokumentáció alapján kell kitölteni.
12. A kitöltés utáni első feladat a funkcionális alábontás teljességének ellenőrzése: amennyiben egy funkció többféle módon is kapcsolódik egy entitáshoz, vagyis több betű is hozzájuk rendelhető, úgy, a funkciót alfunkciókra, folyamatokra kell bontani.
13. Ellenőrizni kell, hogy minden sorban van-e c, h, u vagy d-betű. Ennek megléte garantálja, hogy a funkció végez műveletet valamely entitáson. Az a funkció ugyanis, amely entitást nem hoz létre, valójában nem, vagy csak ellenőrizhetetlen módon végez aktivitást a rendszerben, ezért nincs jelentősége, létjogosultsága. Ezeket a sorokat törölni kell, az FD-t pedig ennek megfelelően karbantartani.
14. Minden oszlop maximum 1 c-betűt tartalmazhat. Amennyiben több helyen található c, úgy a következő esetek fordulhatnak elő:
 - Az entitás szuperentitás, melynek különböző komponenseit kezelik a soraikban c-t tartalmazó egyes funkciók. A szuperentitás elemeit entitásként azonosítani kell és a mátrix oszlopait ennek megfelelően módosítani. (Törölni a szuperentitás oszlopát és hozzáadni az azt alkotó entitások oszlopait). Az ERD-t ennek megfelelően ugyancsak módosítani kell.
 - A c-ket tartalmazó sorok mindegyike olyan funkció, amely ugyanazt a folyamatot, nevezetesen a kérdéses entitást létrehozó általános funkciót tartalmazza. A funkciókat (sorokat) ennek megfelelően módosítani kell, azaz hozzá kell adni a mátrixhoz az új, a kérdéses általános funkciót tartalmazó sort, és módosítani az FD-t.
 - A c-t tartalmazó sor funkciója mégsem hozza létre az adott entitást, a c abban a cellában törlendő.
15. Minden oszlop legalább 1 c-betűt tartalmazzon kivéve, ha bejövő külső entitás oszlopa. A bejövő külső entitásokat a beküldő állítja elő. Ez lehet file, ekkor a rendszer funkciói minden további nélkül felhasználhatják, amennyiben nincs szükség felhasználás előtt

valamilyen transzformációra. Amennyiben transzformációra van szükség (például bizonylat érkezik és azt rögzítik, vagy file érkezik, de transzformálni kell a feldolgozás előtt) úgy az entitást felhasználó (r) funkció létrehozza majd a rendszer működésében szerepet játszó belső entitást. Az az oszlop, amely nem külső, bejövő entitást ábrázol és nincs benne c-betű, törlendő, mivel nem létező, vagy bizonytalan eredetű entitásra hivatkozik.

16. Minden oszlop legalább 1 r- vagy u-betűt tartalmazzon kivéve, ha kimenő külső entitás oszlopa. A kimenő entitásokat a rendszer állítja elő. A kimenő külső entitás lehet egyben belső entitás is, ha azok egymás másolatai, vagy másodpéldányai. (például mentések, bizonylati másodpéldányok). Az az oszlop, amely nem külső, kimenő entitást ábrázol és nincs benne sem r, sem u-betű, törlendő, hiszen olyan entitásra hivatkozik, amelyet semmilyen funkció nem használ fel és adatszolgáltatási kötelezettség sem vonatkozik rá. A szervezet működése szempontjából tehát felesleges.
17. Ha egy sor több c-t és h-t is tartalmaz, érdemes megpróbálkozni alfunkciókra bontásával, mivel nagy a valószínűsége, hogy alfunkciókra nem bontott főfunkcióról van szó. Ideális esetben minden sorban pontosan 1 c, vagy h van.
18. A 3.-7. Lépéseket addig kell ciklusban ismételni, amíg olyan ciklust nem sikerül végrehajtani, amely nem eredményezett változtatást a c/u mátrixban, vagyis a c/u mátrix a 3.-7. feltételeknek egyszerre megfelel.

Ha a c-k száma nagyobb, mint 1 (4.szabály), sor kerülhet általános funkció beágyazására: pl. mind az ambuláns beavatkozás végzése, mind a műtét végzése létrehoz az ellátás során végzett beavatkozást. Ekkor az említett két funkció sorában a c-t h-ra kell cserélni és megjegyezni, hogy a funkciók tartalmazznak egy újonnan létrehozott, általános funkciót.

Funkció/entitás	Beavatkozás		Funkció/entitás	Beavatkozás
Ambuláns beavatkozás végzése	c	→	Ambuláns beavatkozás végzése	h
Műtét végzése	c		Műtét végzése	h
			A/beavatkozás beszúrása	c

A c/u mátrix műveletei során mindig történik olyan módosítás, javítás a funkciók, vagy entitások listáján, vagy kapcsolatán. A funkcionális dekompozíció végső formáját éppen ezért nem is érdemes a c/u-mátrix végső formájának elérése előtt megkísérelni meghatározni[9].

A c/u-mátrix diagonalizálása

A diagonalizálás a funkciók átcsoportosítása, melynek során a c-betűk a mátrix főátlója környékére kerülnek. A diagonalizálás a sorok, majd az oszlopok cseréjével történik.

Az egyes folyamatok időbeni viszonyának csak egy-egy funkción belül van értelme, hiszen a funkciók maguk idő dimenzió nélküliek. Szükséges ugyanakkor csoportosítani a funkciókat azok alkalmazási területei szerint. Ez a csoportosítás a meglévő tapasztalat alapján történik, tehát a szervezet munkafolyamatainak valamilyen szintű ismeretén alapul.

A c/u-mátrix sorait előbb üzleti területek (funkcionális területek) szerint kell csoportosítani, majd az ilyen módon létrehozott csoportok elemeit a csoporton belüli funkciókhoz rendelni, azután a funkciókon belüli alfunkciókhoz (ha van ilyen). Mivel ezek a kisebb halmazok már könnyen átláthatóak, a sorok (folyamatok) funkción belüli sorrendje könnyen megállapítható[32].

Az oszlopok cseréjekor a sorfolytonosan, az oszlopok c-d-h-u szerinti balra zárásával történik.

A felhasználó követelményekben felsorolt funkciók a következő szerint csoportosíthatóak:

Az életciklus szerinti elrendezésbe nem illenek a rendszer-karbantartást szolgáló funkciók, ezekhez a legutolsó üzleti terület (főfunkció) kódjától nagyobb kódot kell rendelni, míg az elrendezésbe szintén nem illő általános funkciókhoz a legnagyobb kódot. Ez által a rendezéskor ezek a funkciók csoportosítva, de a szekvencia funkciók alá kerülnek a c/u-mátrix soraiban (ld. a függelék).

Az oszlopcsere végrehajtása után a speciális funkciók c-betűi a mátrix főátlója felé kerülnek.

A diagonalizálás után 127 funkció (folyamat) és 173 entitás maradt a rendszerben.

Ezek párhuzamos, vagy analóg funkciók, amelyek a szervezet életében elkülönülnek, de folyamataik hasonlóak, az általuk használt entitások pedig megegyeznek. A párhuzamos funkciók ábrázolása az egészségügyi szakellátás, azaz a kórház rendszerében különösen fontos. A járó- és fekvőbeteg ellátástól a modern egészségbiztosítási rendszerek megkövetelik, hogy a beteg egészségügyi dokumentációját az ellátási formától függetlenül egységesen kezeljék. A járó- és fekvőbeteg ellátás dokumentációs rendszere részben kompatibilis, ugyanakkor átjárás van a két rendszer között, hiszen bizonyos járóbeteg ellátásokat fekvőbeteg dokumentációval, és fordítva, kell ellátni a finanszírozási rendszer által előírt adatszolgáltatási kötelezettség miatt. Annak elkerülése, hogy ugyanazon dokumentációt különböző eljárások során ugyanúgy kezelhessék, az általános funkciók beágyazása teszi lehetővé.

Itt is meg kell jegyezni, hogy még a diagonalizálás lépése során is történnek módosítások a funkciók és entitások felsorolásában, illetve a funkciók és entitások kapcsolatában.

4. Eredmények

Üzleti területek kialakítása

A diagonizált c/u-mátrix almátrixai a kórház információrendszerének főfunkcióit, illetve a nyújtott szolgáltatásoknak a rendszer szempontjából optimális struktúrája határozható meg. Ez az üzleti területek, illetve azon belül a funkciók újracsoportosításával történik.

Az üzleti területek kijelölését az üzleti logika szabályai határozzák meg. Ezek:

- Maximális belső kötődés

Az üzleti területen belül (azaz az almátrixban) kevés az üres cella. A üzleti terület funkciói egymással adatcserét folytatnak, vagy ugyanazon entitásokat használják fel. Az adatcsere, a funkciók technológiai sorrendje, vagy éppen ugyanazon erőforrások használata teszi indokolttá a csoportosítást. Az adatbázis elemek és erőforrások optimális csoportosítása idő- és költségmegtakarítással is jár mind a működés, mind a rendszer működtetése során.

- Minimális külső kötődés

A üzleti területeken kívül kevés az u-betű. A üzleti területek közötti adatcserét minimalizálni kell. A törekvés célja nem az egyes területek kommunikációjának redukálása, hanem az adatcsere olyan módon történő átcsoportosítása, hogy az elsősorban a cselevési területeken belül folyjon. A minimális külső kötődés általában nem fogja annak nullára redukálását eredményezni, de nem is ez a cél.

- Logikusan kialakított területek

Nem árt időnként a rendszer tervezőinek figyelmét felhívni arra, hogy nem a megrendelő szervezete van őértük, hanem éppen ők dolgoznak azon, hogy a megrendelő saját tevékenységét az információ technológia eszközeivel hatékonyabbá és hatásosabbá tegye. Éppen ezért az előző két szempont mentén matematikailag szép, de a gyakorlati működés során valamilyen okból használhatatlan rendszerek is kikerülhetnek a tervezés folyamatából, ha az üzleti logika szabályai nem lesznek figyelembe véve.

A fenti szempontok egymás ellen hatnak, így minden esetben a józan ész kompromisszuma szükséges a javasolt megoldás kialakításához. Az egyes szempontok túlsúlya torz rendszereke eredményezt[28]:

- Maximális belső kötődés túlsúlya

Egy sor egy üzleti terület, amit az egyes sorok c betűi határoznak meg. Üres helyek száma üzleti területeken belül: 0. Minden funkció önálló életet él, meg nem értett, magányos zsenikből áll a vállalat.

- Minimális külső kötődés túlsúlya

Az egész, speciális funkciók sorai által meghatározott almátrix egyetlen üzleti terület. Egyetlen betű sem maradt kívül. Minden tevékenységet magába ömlesztő kisgömböc. A hozzá nem értő főnök remekül palástolja tehetségtelenségét.

- Hagyományos üzleti logika túlsúlya

Semmilyen funkció átcsoportosítás nem történik, „minden jó úgy, ahogy van”. Ebben az esetben a rendszer a szervezet életében a hagyományos adatcsere információtechnológiai eszközökkel történő felgyorsítását eredményezi. Ebbe a csapdába a 80-as években, az IBM PC irodai elterjedésével szinte minden vállalat beleesett, amely nem rendelkezett információtechnológiai múlttal. A korszerűtlen és alacsony hatékonyságú szervezet mindent ugyanolyan rosszul tett, mint korábban, de sokkal hatásosabban.

Van értelme a üzleti területeken belül (rész-) üzleti területek kijelölésének, vagy fordítva, adott üzleti területek összevonásának. Ez a rendszer finomszerkezetének leírása. A üzleti területeket kifejező névvel illetve áttekinthetőbbé válik mind a rendszer, mind a szervezet[45][33].

Nem szabad elfelejteni, hogy az általános funkciókkal ki kell egészíteni a üzleti területeket, így az azok felhasználásával újra rajzolt FD-t is. Az, hogy az általános funkciók a üzleti területektől függetlenül, mint a rendszer „utility-jei” lettek ábrázolva, lehetővé teszik azok független fejlesztését, vagy beszerzését a rendszerfejlesztési projekt egyéb fázisaitól függetlenül.

A üzleti területek a szervezetben folyó valamely tevékenységet írják le, ebből következően jól meghatározott szervezeti egység, vagy munkakör(ök) alrendszere, rendszermoduljaként azonosíthatók. A rendszer megvalósítási projektje készítésekor mind a költségek, mind az ütemezés tervezésekor nagyon hasznos kijelölésük: a megvalósítás során nincs értelme a üzleti területeknél kisebb tagolásnak, a megvalósítás sorrendjének meghatározásakor ugyanakkor szerepet játszik, hogy a üzleti terület:

- Mennyi idő alatt megvalósítható.
- Mennyibe kerül.
- Mennyire problémás a szervezet életében.
- Mennyire sürgős.
- Mennyiben szolgálja ki a többi cselekvési területet.

Adatfolyam-diagramok

Az adatfolyam diagramok a funkciók közötti adatáramlást ábrázolják. Az egyes funkciók által kiszolgált munkahelyek, kiszolgálópontok közötti tranzakciószám, a szükséges sávszélesség is modellezhető segítségükkel amellet, hogy a rendszer elemeinek egymással való kapcsolatát mutatják. Az adatfolyam diagram csak a szekvencia funkciókat tartalmazza.

Az adatfolyam diagramok felbontását a földrajzi térképek felbontásának jelöléséhez hasonlóan indexek jelzik: ezek szerint a DFD(0) (Dataflow Diagram 0) az üzleti területek (funkciócsoportok, főfunkciók) közötti adatáramlást, a DFD(1) eggyel nagyobb felbontásban az üzleti területen belüli adatáramlást, s.í.t. ábrázolja.

A diagonalizált c/u-mátrix segítségével az adatfolyam diagramok ábrázolása végtelenül egyszerűvé válik:

- Az üzleti terület, funkció melletti, azon kívül eső lévő r betű a bejövő entitást jelöli.
- Az üzleti terület, funkció alatti vagy feletti, azon kívül eső r betű a kimenő entitást jelöli.
- Az r-et nem tartalmazó oszlopok c-jei külső kimenő entitásokat jelölnek.
- A c-t nem tartalmazó oszlopok r-jei külső bejövő entitást jelölnek.

A fenti meghatározások szerint minden r betű valamely két, azt nem tartalmazó üzleti terület, funkció közötti adatáramlást jelöl.

Látható, hogy a beágyazott általános funkciók bevezetése nem rontja el a c/u->DFD leképezhetőséget, mindössze azt a módosítást kell alkalmazni, hogy azokban az oszlopokban, ahol szerepel h, azt kell az algoritmusban c-ként kezelni és az oszlopban lévő c-t figyelmen kívül kell hagyni.

Adatvédelmi mátrix

Az adatvédelem feladata az adatok elérésének biztosítása az arra illetékes felhasználók felé, ugyanakkor az illetéktelen felhasználók hozzáféréseinek korlátozása, vagy tiltása. A hozzáférés korlátozása[13]. A korlátozás felhasználónként kiterjedhet az elérhetőség időbeniségére (pl. munkaidőre szorítkozva)

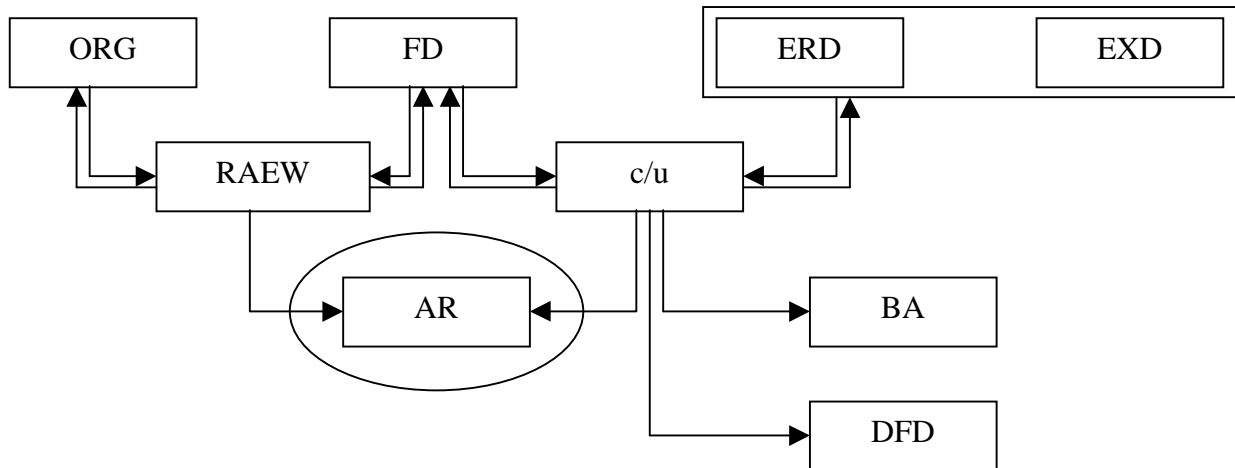
- lokalizációjára (terminálok meghatározása)
- műveletekre (írás, törlés, olvasás, futtatás).

A szervezet organogramja tartalmazza az aktivitást végző beosztásokat. Az egyes hozzáférési jogosultságok az egyes beosztásokhoz (aktorokhoz) vannak rendelve. Az organogram a szervezeti hierarchiát ábrázolja, melyen minden beosztás pontosan egyszer szerepel. A szervezet aktivitását a különböző beosztásokban lévő személyek végzik, kapcsolatukat a RAEW mátrix írja le[27]. E szerint minden egyes aktivitást végző személy beosztásának (szerepének) megfelelően a szervezet egy-egy funkciójához

- R: responsibility, azaz felelősség
- A: authority, azaz felügyelet
- E: expert, azaz szakértő
- W: work, azaz munkavégző

szereppel kapcsolódik.

A Martin-féle szervezési modellen alapuló modell:



A továbbfejlesztett szervezési modell

Jelölésmagyarázat:

ORG (Organogram): szervezeti hierarchia

FD (Functional decomposition): funkcionális dekompozíció

ERD (Entity Relation Diagram): E/K (Entitás-kapcsolat)-modell

EXD (External diagram): külső kapcsolatok

RAEW (Responsibility, Authority, Expertity, Word): felelősség, felügyelet, szakértés, teljesítés mátrix

c/u (create-use): létrehoz-felhasznál mátrix

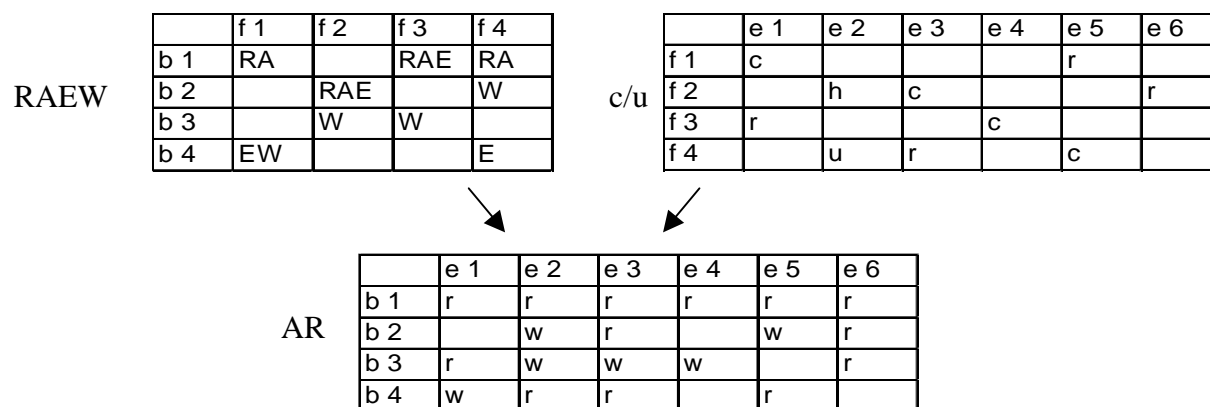
AR (Access Rights): hozzáférési jogosultságok mátrixa

BA (Business areas): üzleti területek

DFD (Data flow diagram): adatfolyam-diagram

Az AR-mátrix a hozzáférési jogosultságokat tartalmazza, amely a szervezeti hierarchia és a szervezet funkciót összekapcsoló RAEW-mátrix, valamint az egyes funkciók és az általuk kezelt entitások kapcsolatát leíró c/u-mátrix szorzataként jön létre. A RAEW-mátrix szerint minden funkcióhoz 4 beosztás kapcsolódik, pontosan egy R, A, E, és W szereppel, ezen funkciók pedig 5 módon kezelnek egy-egy entitást: c, h, r, u, d. Ezek szerint egy beosztás egy-egy entitást 20+1 módon kezel (a +1, ha sehogyan sem). Ezt a bonyolult kapcsolatrendszert automatikusan kezeli az itt ábrázolt modell:

A RAEW w-je írása a c/u betűjét engedélyezi az AR-ben, míg minden más RAEW-beli betű olvasási jogot a c/u-szerinti kapcsolatban.



Hozzáférési mátrix

5. Összefoglalás

Jelen értekezés a kórházi információrendszer követelménydokumentációja meghatározásának folyamatát írja le a kórházi igények felmerülésétől a szoftverspecifikációhoz szükséges részletek meghatározásáig. Rövid áttekintést ad a hazai kórházi informatika evolúciójáról, rámutatva arra, hogy a követelménytervezési folyamat mellőzése az ilyen rendszerek fejlesztésének rendkívül alacsony hatékonyságát és hatásosságát eredményezi.

Az értekezés első részében rövid áttekintést ad a követelménydokumentáció szerepéről és fajtáiról, valamint a követelménytípusokról.

Az értekezés második része egy konkrét kórházi információrendszer követelménydokumentációjának elkészítését írja le lépésenként, s e konkrét rendszer tervezésének lépései során tartalmazza a tervezési módszer továbbfejlesztését jelentő tudományos eredményeket. A kórházi igények megfogalmazása után meghatározásra kerülnek a felhasználói követelmények. A felhasználó követelmények jellemzően nem térnek ki a nem funkcionális követelményekre, mivel a megrendelő egyrészt nem tudja kifejezni saját igényeit, másrészt magától értetődőnek tartja a használhatósági kritériumokat. A szakterületi követelmények jellemzően a másodlagos, a nem gyógyító-ápoló tevékenységgel kapcsolatban jelennek meg.

A rendszerkövetelmények meghatározásának lépéseit az intézet feladatrendszerének meghatározása kezdi, mivel a rendszer használhatósága elsősorban a rendszertől elvárt mérésekkel kapcsolatos adatszolgáltatással függ össze. A funkcionális dekompozíció, vagy

funkcionális elemzés a rendszer funkcionális követelményeit főfunkció-funkció-folyamat bontásban részletezi. A kezelt entitások meghatározása a ilyen módon meghatározott funkciókhoz történő hozzárendeléssel történik, azaz minden egyes funkció vagy folyamat esetén meg kell határozni, hogy milyen input paraméterek szükségesek a funkció működéséhez és milyen outputokat produkál a kérdéses funkció vagy folyamat.

A módszer fókuszában a c/u-mátrix áll. Ezzel a funkciók és az entitások kapcsolata lesz szabályozva. A mátrix soraiban a rendszer funkcióit, míg oszlopaiban a rendszer által kezelt entitásokat elhelyezve az összes lehetséges funkció-entitás kapcsolat könnyen áttekinthető és meghatározható. A mátrix műveletei során a rendszer teljessé és ellentmondásmentessé válik, majd a diagonalizálás során a rendszer architektúráját, adatfolyam diagramját is meghatározza. A mátrix műveletek az entitásokra is hatással van, így a végleges adatmodell és a külső kapcsolatok is a c/u mátrix végleges formája után határozható meg.

E módszer nem terjedt el a hazai gyakorlatban, aminek oka a következő két probléma:

1. A diagonalizálás során a funkciók sorrendjének meghatározására két módszer ismeretes a szakirodalomban: ezek az életciklus szerinti sorrendiség, illetve a funkció-függőség szerinti sorrendiség. A funkció-függőség szerinti sorrendiség egyszerű, például gyártási folyamatokban jól használható, de a visszacsatolásos rendszerekben (így az egészségügyben) nem jelent megoldást. Az életciklus szerinti sorrend meghatározásakor problémát okoz, hogy bizonyos funkciók nem illeszthetők be az életciklus szerinti időrendi sorrendbe, vagy esetleg több helyen is beilleszthetők lennének.
2. A funkciók és entitások a c/u mátrixban négyféle kapcsolatban lehetnek. Ezek jelölése: „c”, azaz létrehoz, „u” azaz felhasznál (módosít), „r” azaz olvas és „d” azaz töröl. Egy entitáshoz csak egyetlen funkcióval lehet „létrehoz” kapcsolata, vagyis kizárólagos jogot kell adni az entitásokat létrehozó funkcióknak. Ennek megoldása gondot okoz a párhuzamos, vagy konkurens funkciók esetén; például egy betegnek a járó- és fekvőbeteg szakellátás során végezhetnek ugyanolyan vizsgálatot, a különböző formájú ellátási eseteknél ugyanazon entitásokat, jelen esetben a vizsgálatot mindkét ellátási protokoll szerint rögzíteni kell. A vizsgálati eredmények rögzítése szigorú protokoll szerint, a funkciók időrendiségében meghatározott mindkét ellátási forma esetén. A „vizsgálati eredmények rögzítése” funkciónak tehát az életciklus-szekvencia szerint mind a járó-, mind a fekvőbeteg ellátásban szerepelnie kell, ami önmagában nem okoz gondot, hiszen átnevezéssel egyértelművé tehető, hogy melyik funkcióról van szó, pl. „járóbeteg ellátás során vizsgálati eredmények rögzítése” és „fekvőbeteg ellátás során vizsgálati

eredmények rögzítése”. A gond továbbra is az, hogy mindkét funkció ugyanazt az entitást hozná létre, márpedig a c/u-mátrixban egy entitás sorában csak egyetlen „c” szerepelhet.

A két probléma együttes megoldása a nem szekvencia-, vagy általános funkciók bevezetése. Az újonnan bevezetett általános funkció tartalmazza az entitást létrehozó kizárólagos jogosultságot. Kívül esik a szekvenciákon, de a szekvencia funkciókból történő meghívással az entitás létrehozásának funkciója beágyazódik a hívó eljárásba. A párhuzamos funkciók ez által ellentmondás mentesen kezelhetőek. A fenti példa esetén az újonnan bevezetett „vizsgálati eredmény beszúrása” funkció tartalmazza a „c” (create), míg a két eredeti funkcióhoz „h” (have create) kapcsolat rendelhető, ami mutatja, hogy a kérdéses funkció beágyazott funkcióként tartalmazni fog egy általános (nem szekvencia-) funkciót, és ezt majd a funkcionális dekompozícióban is így kell feltüntetni.

Az új funkció-entitás kapcsolat bevezetése mellett továbbra is fennáll a c/u-mátrixnak az adatfolyam-diagramra történő leképezhetősége, azaz nem romlik ez a függvény. A leképezésben annyi változtatást kell mindössze alkalmazni, hogy azokban az oszlopokban, ahol van „h”, azokra kell alkalmazni a „c”-re vonatkozó szabályt.

Az értekezés tartalmazza a rendszer üzleti területeinek kialakítási szabályait, illetve be is mutatja azok alkalmazását. Tartalmazza kész rendszer funkcionális dekompozícióját a szekvencia- és nem szekvencia funkciókkal, illetve rendszerfunkciónak nevezett funkciók csoportjával, amelyek az előző két csoport egyikébe sem illenek, mivel sem szekvencia, sem beágyazható általános funkciók. Ábrázolja az értekezés a rendszer külső kapcsolatait leíró diagramot, valamint a teljes rendszer adatmodelljét.

A dolgozat bemutatja, hogyan készíthető el egyszerűen a teljes rendszer adatvédelmi mátrixa. Ez szerves része magának a szervezési modellnek, így bármilyen adatszolgáltatási követelmény, funkcionális változás vagy a szervezeti felépítés átalakítása automatikusan maga után vonja az adatvédelmi mátrix aktualizálását.

Az értekezés végül tartalmazza a modell kórházi információrendszer nem funkcionális követelményeit.

The model of the (hospital) information system

6. Premise

The method of planning the requirements

The dissertation gives a methodology to the phase of plan and design the requirement documentation of hospital information systems, applying the theory on a given system.

The evolution of the hospital information systems also in Hungary began in the 80's as implementing applications from other spheres of the life, such as general ledger, warehousing, waging. The register programs were the ancestors of the health applications. They were able to make at least a snapshot, but it was unsuccessful to function them continuously, since the unconcern changed the enthusiasm in the beginnings. Contrary to the taxing and financing in the economy, there were not existing registering and reporting disciplines in the health area. This was the reason, that developing of the applications of the curing was only produced by the co-operation of some physician and engineer or mathematician, who knew the computer.

Thanks to the widespread of the personal computers and their systems, the easy-to-settle and cheap database-managers in all medical areas computers appeared. These – as due to the lack of regulations on how to implement computer-system in hospitals it was depending on the decision of the head of the hospital – were far from being solid neither in point of software platform, nor the function of the application, nor the handling of the document. To do the same tasks, similar programs were made; very often in one institute there were several types of programmes used for the same profile in different departments. The requirements, regarding the endurance and the quality of these systems were low, as it was still the paper-based documentation, which was in effect primarily, computer documentation was only collateral.

The change was brought by introducing the performance-principled financing system [5][47][34]. This system assigned a financial base-document to the curative work and assessed the hospitals an order of supply of data. When the government issued the order of supply of data, it provided the hospitals with PCs and software for free of charge. Even today there is a hospital, which does not have any other healthcare application than that software. Part of the hospitals took measures to introduce integrated hospital information systems; these systems were mainly adaptations of systems developed abroad. According the present situation systems that are able to attend central management, the healing processes and the decision support in connection with curative work as well, are rare to find in hospitals. Many institutes made serious efforts to introduce systems - which had been operating successfully

elsewhere - without striking success. Systems don't meet the general sympathy of the organisation, the main requirements for them get simplified to the level of ensuring the data-providing service, which is necessary for the performance-principled system and last but not least to make the hospital management concerned.

Applying information technology didn't bring efficiency improvement in its train. Measuring and proving it is simple: the numbers of complications, the hospital time, but even the institutional by-results-analyses don't show institutes using information systems more effective than those who totally omit that. The reason for this is omitting a very important phase in the process of implementing a system, namely defining the requirements of the system based on professional and realistic demands [11]. The formation of this situation was necessary: the institutions don't have the essential and actual specialized knowledge and/or the capacity of professionals to be able to define the requirements, and the supplier of the systems are not interested in that. The supplier's aim is to make the product at the smallest cost, that is, to implement it at more places with minimal changes, if any. But this, due to the different needs and requirements (see later, for instance, folklore) is uncomfortable to use, so sooner or later it leads to implementing systems with unused functionality.

Defining the requirements expected from the system it is of fundamental importance to define the data model, which serves as the base for defining the database of the transaction system. Another very important step is to define the functions expected from the system, which really do serve the practical work processes [10]. At the area of economy, practice is done by well-defined method protocol, mainly based on external circumstances. The working processes and lists of curing are not standardised, so only after the thorough studying of the concrete activities can the requirements of the system be defined. Those institutions, which have some sort of quality control system, have the upper hand in this case: not necessarily because the regulation is adequate but because the part of the documentation, which is necessary to define the requirements, are available. Compiling the requirement-documentation also makes the quality system change, this is what the hospitals are most unwilling to do so. They are hard to understand that introducing the informational system means an intervention into the daily work processes and also into the documenting process, therefore it is an intervention into the quality control system, too, which is describing that.

This dissertation is describing the process of defining the requirement documentation of the information system of the hospitals, from the emerging of the needs to the definition of specifications of softwares. It is introducing the document called User's Demand that is the antecedent of the document called User's Requirements. After the data model and the function-analyses, the functions of the system and the relation between the managed entities

are defined by the attributes of the c/u matrix¹ which is described in the Martin-model [30][35]. In the thesis the combination of function dependency and the life-cycle model define the sequence of the functions. Applying this model on simple work process is easy by the P.C.I.³ organising model but it is unable to handle the concurrent processes in case of systems with great complexity. In these cases it often leads to the following contradictional result: creating the same entity has to be done by several functions which is forbidden in system models, as this is the main cause for the transaction systems to crash down. Since the application of the function-dependency model with systems providing curing-healing activities is not realistic, the solution for the problem is the solution of the problems with concurrent processes in the life-cycle model. In order to solve these problems, this dissertation is introducing the non-sequence functions that are parts of the system but not necessarily parts of any processes by the life-cycle model. When filling in the c/u-matrix it solves the problems with the competitive processes by assigning “c” to the relation of this function-entity and at the same time, introduces non-sequence functions marked with “h”.

After the theoretical scaling of the system requirements documentation, the dissertation presents the documentation of the concrete system requirements of information systems based on real demands by real hospitals according to the structure exposed, also, it is demonstrating the applicability and the efficiency of the above-mentioned solution.

7. Literature antecedents

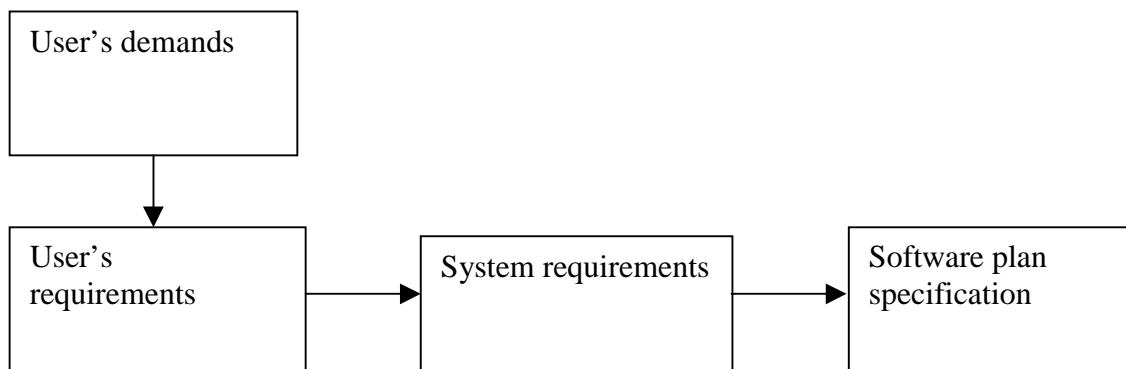
The method of requirements planning

The requirements of an information system are usually very complex. The problems to be solved are hard to understand, and due to the different needs and demands of the different users, or other requirements concerning the system, the expected way of running it, is not evident. The services to be provided by the system and the description of its restrictions are called *the requirements of the system*, and the process of inventing, analysing, documenting and controlling of these restrictions and services are called *requirements-planning* [44]. If a company (procurer) would like to sign a contract on a great software-developing project, then, it has to define its demands in quite an abstract way so that the solution would not be pre-given. Requirements have to be defined in a way that more suppliers can make their quotations, in this way offering more options for the procurer to satisfy its needs. As soon as one of the suppliers wins the contract, it has to write an even more detailed system definition

¹ Some sources refer to it as CRUD matrix

² Process Control Information System

for the procurer so that it can fully understand and approve what the software is going to do. Both documents can be called the documents of the system requirements.



The types of the requirements documents:

- The *User demands* are the procurer or user's claims for the information system services, although it is often and full of contradictions. We should not expect correct and professional definitions of the user's requirements, because the user is not a professional in information technology. In the case of several potential users, each of them will mark its own points of view. This conflict will be solved during the making of the *User requirements*.
- The *User requirements* documentation: containing expressions, what services have to be performed by the system and the restrictions are. These are natural – or spoken - language expressions, containing figures, drawings, tables, drafts and patterns. *It made by the user/procurer, and usually takes very beginning in the developing cycle.*
- The *System requirements*, which particularise the system services and system restrictions, are also called *functional specifications*. The contract on the system providing between the procurer and the supplier should be based on this documentation. *The provider, considering the user requirements, makes it.*
- The *Software specifications* are the abstract description of the software plan. It is the basis of the particular system plan and implementation, which is needed to plan the additional hardware and software elements. That can be adding further details to the system requirements. It contains both the system user requirements and the detailed system requirements. The user requirements and the systems requirements can be within the same documentation, but then the user requirements can be cited as the introduction of the system requirements.

For example, the software requirement specification should perform the following assumptions:

- Determine only external behaviour.
- Specify any restrictions that can prevent a successful implementation.
- Must be easily updated.
- It can be used as reference for information system maintenance.
- It should contain forecasts for the lifecycles of the information system.
- Whenever possible predict and offer solutions to unwanted events.

Requirements and restrictions

The requirements can relate on the claimed particular services, the method of the consolidation or properties of the whole system or its individual component, such as quality or performance parameters. The classifications of these categories are the followings:

Functional requirements

These describe the supported services and functions. It contains the behaviour of the system under several situations.

Theoretically, the system functional specifications must be complete and free of contradictions. Completeness means that the entire user defined service must be contained. Free of contradictions means, that there must not be any difference of opinion at the requirements level. In practice, it is very hard to get the completeness and requirements free of inconsistencies. It's reason is partially due to the inside complexity of the systems, while on the other hand several viewpoints mean different opinions or ways of thinking about claims. These inconsistencies do not become obvious immediately.

To work out the completeness and to minimise inconsistencies of the functional requirement specifications, there is well-used method that describes the function-entity context (the c/u matrix). The functional decomposition is the way to determine how the system service decomposes into processes. The entity of the system will be ordered on these functions and processes before determining the system's data model.

Non-functional requirements

These are restrictions on the system's functions and services, or on the whole system. They include time restrictions, constrains on the development process, standards, etc. These are not in direct contact with the system-supported functionality. They can be related to system properties, such as reliability, response time, storage capacity claims, configuration, requirements to individual system components (such as I/O capacity) or description of data interfaces.

A number of non-functional requirements relate to the entire system, not individual components. These are usually more important, than the individual functional requirements, since an unsuccessful performance of an individual requirement can slack the system, while the failure of a non-functional system requirement can make the whole system unusable.

A hospital information system, using non-adequate unit to store and forward the result of diagnostic examinations, gives unusable data to the physician, instead of the needed ones. Obviously, this system is unavailable to support the healing.

Similarly, if it does not perform the financial data supply in the form given by the recipient, it is not able to document the hospital income, which is the most basic requirement.

The non-functional requirements are not definitely related to the actual software system. It can determine for example, that the requirement documentation and software specification must be written according to the rules and structures of a named model or methodology, and create and maintain the UML standards. It can reference the prescriptions of applied quality assurance system, internal regulations or external laws or other circumstances. They can be special user claims, algorithms or data handling commissions.

Domain requirements

These are mirroring the properties of the speciality, the knowledge of the special domain. These domain requirements can be either functional or non-functional. Their origins are the rules, algorithms and constraints of the speciality, instead of the user claims. These requirements contain the basic conditions of the availability.

Domain requirement of the dialysis treatment

The Kt/V value is an indicator of the dialysis treatment. It correlates to the expectation of life.

$$Kt/V = -\ln\left(\frac{CN_2}{CN_1} - 0,03\right) + \left(4 - 3,5 * \frac{CN_2}{CN_1}\right) * \left(\frac{UF}{BW}\right),$$

where CN_1 is the content of the urea-nitrogen before treatment

CN_2 is the content of the urea-nitrogen after treatment

UF is the ultra filtration of the given treatment, given in liter

BW is the body mass after the treatment, given in kilogram

The domain requirements are expressed on the special area's language, which is very hard to understand by the software designers. Specialists of the domain can simply leave out information of the requirement, since those are evident for them. On the other hand, there may not so evident for the system designers. This is the reason of using computing specialists

besides the system designers, who are experienced the given speciality, both the healthcare and the information systems [11].

8. The model of the hospital information system

The model presented in the dissertation is based on requirements and demands of Hungarian hospitals. These omit references to concrete hospital surroundings; on the other hand they formulate user's demands of typical Hungarian circumstances. The enumeration of the expected qualities of the system was done with the cooperation of experts of medical, economical and the experts of the information technology of the practising hospital [14][51]. The rest of the thesis is describing the requirement documentation of the (sample-) system containing the steps of defining the system requirements. Each passage begins with an introduction and a review of the method, then, after writing down the concrete data and results, it ends in an evaluation and if necessary, explanations.

The expectations of the hospitals for the system

The first stage of planning the system is to define the demand. Compiling the information technology demands of different specialties is formulated in natural languages. It should be borne in mind that hospitals are not software developing companies; therefore they cannot be expected to write their demands in the language of information technology.

When the demands are written in natural language, some problems may occur, in order to avoid them, attention must be paid to the following factors:

- *Lack of being unambiguous*

Sometimes it is hard to use a language properly, evidently, without making the text diffuse and difficult to read.

- *Requirements get mixed up*

Functional requirements, non-functional requirements, the aim of the system and planning information are not always separated clearly.

- *Requirements get combined*

Different requirements can be defined as one single requirement.

Services demanded

The user's demands describe the functional and non-functional requirements in a way that they should be easy to comprehend without detailed technical knowledge. They describe the expected running of the system, omitting internal functioning. The user's requirements is not recommended to be defined with the help of the implementation model but it has to be written in a natural language, by using blanks, and easy and simply-to-understand diagrams. When planning the user's environment it is worth drawing in experts or a professional

company, which is up-to-date regarding information technology, which is usually beyond either the capacity or the preparedness of the hospital experts. [11]. Experts like these prove useful in the phase of the call for a tender, although in many cases there is a suspicion that they are the exploratory of a supplier.

Many procurers make the mistake of trying to define the internal functioning of the system in too deep in details. With this, not only don't they remove the responsibility from the supplier in case of not satisfactory running of the system, but their own knowledge – compared to a software and hardware developer's knowledge – is probably limited, and in this way they hinder their own possibilities. Also, it can happen, that defining the "how" creates such a contradiction in the system requirements that leads to the impracticability of the non-functional requirements. Of course, there may be exceptions such as requirements of a special area, the external requirements, or the regulations concerning the documentation and the language of the system when it is impossible to ignore the intervention into the process of the planning.

Analysing the user's demands and making them clearer leads to the definition of the services demanded by the user [10].

System requirements

System requirements are detailed descriptions based on user's requirements that are the base of the communication – about the system - between the user and the supplier. This is the object of the contract of realisation; therefore it must contain the whole system in its coherency. The base of the software specification document, the starting point of the system plan for the software designers. It may contain the different models of the system, for instance an object or a data-flow model. System requirements must describe WHAT the system has to do and not HOW it should be achieved.

Tools for defining system requirements

The user's requirements – by its name – contains the user's (procurer) requirements. Its is far from being complete or free from contradictions, which is primarily, the consequence of the user's subjective value judgement and his knowledge of the system. System requirements, though, must be complete and free from contradictions as partly this is the base for the contract of realisation, and partly because it is also the base for the software plan specification. The supplier (developer) therefore must examine in details the external functioning required from the system and the conditions of implementability.

Description of the task system of the organization to be provided

The system to be delivered must provide the procurer's (part-) activity. The activity to be provided cannot be abstracted from the organization which is carrying on that activity, since some concrete activities of the organization - which are applying the system - are defined mainly by the system itself, and also it should be borne in mind that the system is inspired to provide an organization of well-defined task system. The task system must be revised and updated before planning the information system [45]:

8. Mission – defined by owner. Brief, high-level wording of the intentions and policy of the organization that is applying the system. (Who am I? Why am I?)
9. Strategy – Drafts of policies and plans, which defines the operating of the system in a given period (without well-defined beginning or end). The environment and the mission of the organization define it. Strategy is always defined by the management. (How do I do it? What market do I target? What means do I use?)

Under the title “environment” all factors must be listed which have an effect on the functioning, since their existence, their qualities and changes influence the system elementarily and on the merits. Factors like this:

- Political environment (Predictability, legal certainty, corruption, stability).
- Infrastructural environment (telecommunication, logistics, technical conditions).
- Legal environment (laws, other (for instance, local) regulations)
- Resources/Means (technological, financial, material, human).

10. Objectives, Vision – A general definition of those directions that define the work done on certain areas without defining a concrete date. Vision of future, where strategy leads. (What direction do I wish to go?)

11. Goals – Defining concrete goals, which are commensurable and have a concrete deadline. These serve for measuring whether the organization is on its way to reaching its objectives.

Among goals we can distinguish between constant or continuous, the tactic or short term and the strategic or long-term goals. It can also happen that part of the tactic goals can be interpreted as specifying strategic goals, while strategic goals can be the phrasing in points of the mission itself. Constant goals mean goals that have to be achieved over and over again. In case of hospitals goals like this are the effectiveness and the efficiency of medical attendance.

12. Critical success factors(csf's) – factors of key importance that are essential to achieve goals and the objectives and to accomplish the mission. These factors are commensurable, they are indicators. Critical success factors must be classified by their seriousness,

therefore distinguish between “critical”, “important” and “advantageous” success factors. This classification is a good method to show the differences between the fulfilling the success factor and the necessary expenditures [22].

13. Measuring – Controlling goals and success factors, carrying out systematic and regular measurements. (for instance: achieve a goal at given date) Systematic: measuring has to be carried out the same way all the time. Regular: not spontaneous, but it takes place at a planned point of time.
14. Problems – Problems can emerge as a result of measuring or in case of not achieving the goals. One has to prepare for these problems, has to be able to handle them. Problem doesn't necessarily mean something bad. There may be external and internal problems. External problems are caused by the circumstances out of the organization; such as the changing of the mission, or the changes in the environment. Handling the external problems usually go together with the changing of the strategy and goals. (alter profile, budget). The causes of the internal problems are to be found in the organization, the system is expected to play an active role in the preparation for the decision in case of both external and internal problems.

Preparations for the availability of the system, preparations for the planned and the non-planned shutdowns and making catastrophe-plans are all in the group of problems.

System requirements must contain the functions, which enable the measuring to be carried out, as they are necessary for the user to achieve its overall goals. The user's demand usually refer to these measurements, but due to the lack of their concrete formulating, they are usually left out of the user's requirements [12].

Note

Each measuring means to create one or more entities. The list of services expected from the system must be expanded with measurements.

According to the paragraphs above, the process of requirements planning should always begin with a feasibility study. This study must contain the task system of the procurer, in this case, the task system of the hospital. Thereby, that the supplied information system is preceded from the task system, the utility of the system in different organisation is easy to test. Inasmuch as the task system of the new organization is different form that of the concrete organization, the difference between the actual and the necessary system can be defined, as well.

The aim of the same phase is to reach the definition of the critical success factors that are essential for the running of the system, and also to define the entities and functions, which are essential to measure that. Measurements will be the reports of the information system. It

may be conceivable, that the procurer doesn't request the measuring of certain success factors, or it can also happen that the procurer doesn't even want a single report like this to be supplied with. Still, the supplier must define them, since the system requirement may not only contain the realised but the excluded functions, as well, i.e. what functions are supported by the system and what are not, it is improving the completeness and the unambiguity of the document.

Functional decomposition (FD)

The functions that are defined during the definition of functional requirements can be divided into function groups of primary and secondary main functions. The primary main functions are the functions that define the activity of the organization, and the functions that are typical of the profile of the organization, while the secondary main functions are the rest that don't belong to the group of the primary main functions. (This division is useful in aspect of reusing the extant systems and the communication with other systems: to realise the secondary main functions a completely different systems of other organizations can be used, too, while to realise primary main functions only the systems of the organizations with the same profiles can be used.)

The functional decomposition specifies the activities of the organization by [29]:

- Main function (or function group): services in relation with a certain activity-area.
- Function: concrete service within the main function. The function and the main function is describing a constant, extant service, but not a concrete activity. The function may be divided into further functions.
- Process: A concrete activity that can be described as part of a function. Process is an activity made up from procedures, in its input and output there is a well-defined activity, and it typically has – unlike functions – a beginning and an ending.

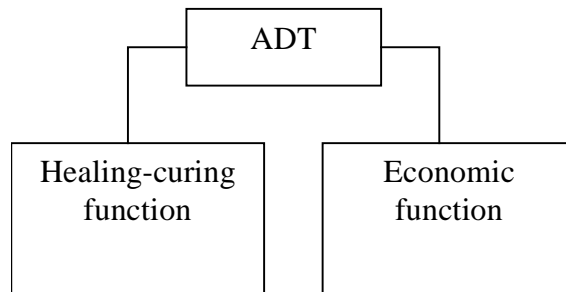
(Henceforth, all 3 services will act as functions when it is not necessary to differentiate amongst them.)

This enumeration is specifying what services are brought by the given function but does not touch upon how that is done. The software plan specification will contain this latter, which is describing the procedures making up processes [20][41][53].

The enumerated functions in the user's requirements illustrated by the functional decomposition, the occurrent parallelisms, and also the inadequately defined functions, namely those, which are not decomposed by processes, become clear. The nucleus of the decomposition is: demanding a service can only be achieved by defining it concretely and in a detailed way, that is, decomposing it by procedures. When planning system requirements the

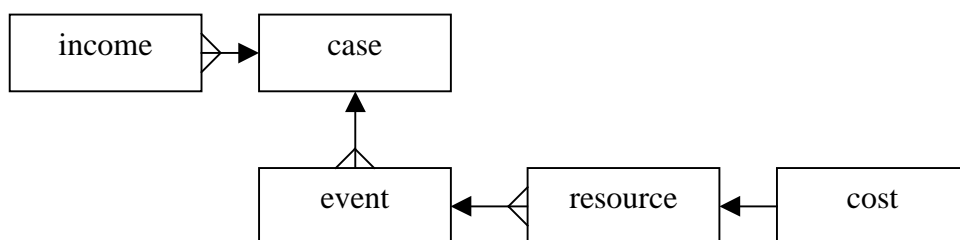
missing itemizing has to be carried out, or the demanded service has to be deleted from the functions of the system.

The architecture model of the hospital system is the model of the economic organization doing healing-curing activity:



The data-supplying system of the medical attendance provides the documentation of concrete sustenance cases, and the decision-support during the process of medical attendance. The resources used during the healing-curing activity have to be assigned to the sustenance cases by the rules of the medical and caring documentation. Parallel with this, from the economic point of view, certain sustenance cases that are documented according to the financial system must be regarded as a financial source document, that is, the sustenance case can be identified with the income by the rules of the financial system. With this income the cost of the resources used during the case must be contrasted. These costs by the rules of economy are direct or indirect costs and to be accounted

The simplified data model of the medical attendance:



One way of interpreting the illustration above:

- „To one case, 0 or more events belong to”.
- „To one resource, 0 or one costs belong to”.

The detailed explanation of the illustration can be found in the data model chapter 3.2.2.4.

The events (documentation by doctors and nursing staff) which are in relation with the case belong to the ADT system (Admission, Discharge and Transfer of patients) which provides the data managing of the sustenance cases, on the other hand, the allowance which was received for the sustenance, and the cost of the directly, or indirectly used resources can be assigned to the case.

The architecture of the system can be distributed to a module that provides the data managing of the medical attendance, and another module that supports the economy of the hospital and a sub-system that integrates these modules.

Note

In aware of the user's requirements the supplier's job is to define more detailed system-requirements by a base-model (conception). The hierarchic build-up of the organization of the procurer has to be written down and the scaling of the task system of the company.

The procurer might think these tasks are useless, or can even be resentful. The developer (supplier) mustn't forget that the procurer's failure in using the system is the supplier's failure, as well. In all cases when the procurer wants to stand off a phase of the development, it is the supplier's interest to document correctly the circumstances for leaving out that phase.

It is prominent that the services in relation with economy act in a detailed way, and are adequately grouped according to the build-up of the actual organization; on the other hand, the services in relation with medical attendance act only in rough. It bodes the putting the secondary functions forward, that is showing the Parkinson-like symptoms of over-bureaucratising.

It has a historic cause. The application of information technology in the Hungarian and foreign health care started with ordering the liability of reports for the financial and accounting systems. This external effect made it obligate for the hospitals to use the electronic data managing irrespective of its thrift. Such a constraint didn't emerge in area of medical attendance, therefore that area is more immature. The quality criteria of the health care are still not established, therefore the restraint in order to be in accordance with it is not effecting the application of information technology. There are two areas where this restraint of application came up: the performance supply of data, which is essential to create the vast majority of the income which is essential for the running of a company (its implementation still clashes with the heavy opposition of the provider staff), the other one is the electronic medical report which is reducing the administration of the medical staff.

Defining the handled entities

In order to define the system requirements it is important to enumerate the entities that are handled by the system.

It is worth representing at each entity where that is from (external entities, or produced by a function) and what it is used for (external entities, or it is essential for a function to operate).

The converse method is useful, too, that is, in case of certain functions it is useful to enumerate those entities that the function is using or producing.

The external entities are not really parts of the data model – they are to be enumerated by writing down the user's demands – but they are great for controlling: all attributes of the external entities have to act as an attribute of a certain internal entity in the data model. External entities are data that make communication (out of the system) available: they can be incoming or outgoing entities. A part of them describes the dialogue between the procurer's organization and the outside world, while there are entities that are out of the system (external-external: EE) but provide the data exchange within the organization. (external-internal: EI). The data exchange between authorities, customers and suppliers belong to the EE group while other entities that realise the data exchange between other sub-systems of the organization, for example realising the back-up functions. It is worth differentiating between the two groups.

The revision of the external-internal entities is recommended to be carried out regularly. The outgoing external-internal entities have to be marked whether they are going to be put in the back-up or to a well-defined user within the organization. This latter has to be marked on the organogram of the organization. The data to be backed-up is described by the maintained back-up system. The destination and the necessity of the external-internal entities that end up at the user must be revised regularly. It often happens that a new formal or content requirement is ordered to these entities while the previously used one is not overruled.

The base document of the out-patient case is the Ambulant Data Format. 6 years after the introduction of this Ambulant Data Format, there was still a hospital, which used the previously used Ambulant Blotter, as well, since they still had some piled up in their warehouse. The warehouse, as the blotters were continuously in use, kept re-ordering them. No one seemed to be bothered in the hospital by the fact that the Ambulant Blotter didn't need to be sent to anywhere since the Health Insurance disposed only the Ambulant Data Format

The external-external entities contain formal and content restrictions that are defined within the frames of a law or a treaty/agreement, so there is no possibility to change them. The external-internal entities are defined within the organization by rules of business, quality control or production, therefore to change them in case of extra services provided by the system is a question of decision within the procurers organization. It is worth noting which functions produce the outgoing external entities and which functions use the incoming ones [2]. The external diagram is a great tool for the graphic illustration of it:

Defining internal entities by the user starts with enumerating the entities that had been used before implementing the system. The entities that are necessary to carry out measuring following from the task system, define further entities and those that enumeration of the outputs and inputs of the functions which are enumerated in the functional requirements.

It is worth noting that the functions and the procedures of the system are also functions. In this chart, all entities including external ones can be found. External and internal entities are not disjunctive aggregations, that is, an entity is not an external or internal entity, but it can be both external and internal entity, as well.

The relation between the functions and the entities, the c/u matrix

To an organization, the implementing of a new information system necessarily means the changing of the actual array of work and the traditional distribution of tasks. Implementing the system means re-grouping functions that is to say, it means the appearance of new tasks and new functions, but at the same time it means that other functions cease to exist. If this is not so, to achieve a more efficient running becomes pending. The operations of the c/u matrix carry out the revision of reforming the functions and the entities of the organization and carry out the optimal regrouping and reforming in the aspect of the organization.

The c/u matrix is the most important element of the planning model, illustrating the relation between the functions and the entities of the organization. In the rows of the matrix, the functions can be found, while in the columns, the entities are. [36][30].

When filling in the rows of the matrix, it can be assumed that the functions had been detailed in the depth of process. It is not a big fault, though, if not, because the following operations will show those functions that had not been detailed satisfactorily. 2 kinds of process can be differentiated:

- Specific, or sequence-process: within the function, it is typical to have a position in time compared to other processes. An example for this is delivering goods from stock, which can only happen before its remittance, or setting up a diagnose for a patient which can only happen before ordering the therapy. Setting the sequential order is carried out by the life-cycle principle, that is, the real-life, or a process-cycle.

- General, or imbedded process: it may appear in several functions, it cannot be positioned evidently in the sequence of the life-cycle model. These processes will be used as parts of the specific processes, imbedded into them. An example for this is carrying out an intervention on the patient, which takes place during either in-patient, or out-patient sustenance, which are well differentiated. The general functions can be easily differentiated by their names beginning with “G”.

The first operation of the c/u matrix is to filling it in which is based on the sub-chart of function-entity.

The possible relations between functions and entities:

- The function only uses the entity but does not carry out an operation on it: r (read)
- The function creates the entity: c (create)
- The function uses and modifies the entity: u (use)
- The function deletes the entity: d (delete)
- The function directly creates the entity, having the general functions of creating an entity: h (have create)

The method to validate the c/u-matrix

1. Fulfil the c/u-matrix by the user requirements, or if it exists, the system requirements.
2. Check the rows, that is if a function associates to an entity several ways, that is there are more letters in an intersection of the function’s row and the entity’s column, the function must be decompose at least the number of function as the letter were shown before.
3. Check that every row must contain one of the “c”, “h”, “u” or “d” letters. It guarantees the function to make something the entity. The function, which does not create, has create, delete or changes the entities, does not do any activity in the system, or it makes that without any control. These rows must be deleted, and also have to maintenance the functional decomposition.
4. Every column must include maximum 1 of “c”s. In the case, that the column includes more than 1 “c”s, one of the following cases can occur:
 - The current entity is so-called super-entity, where several components are handled by several functions. There must be identified these component as entities, delete the column of the super-entity and add the identified entities columns to the c/u-matrix. The ERD must be corresponded these changes.
 - The rows, containing the “c”-s, embed the same non-sequential (or general) function. The rows must be changed, that is change the “c”s to “h”s, and add the non-sequential function’s row the c/u-matrix, containing “c” in the current entity’s intersection.

- Some “c”s are badly signed in the matrix, the wrong letters must be deleted or changed.
5. Every column has to contain at least 1 “c”, except, if the entity is an incoming external entity. The sender creates these entities. These can be file, in that case the system’s functions can use it without change their form. In the case, any transformation is needed (for example, a paper based documentation arrives and it must be recorded, or an incoming file must be converted) the function, which reads (“r”) the entity, creates (or has create) the entity, which has role in the system’s functionality. The column, which does not describe incoming external entity and does not contain any “c” letters, must be deleted, because its origin is unknown.
 6. Every column must contain at last “u” or “r” except, if the entity is outgoing external entity. These entities are created by the system. The external entity can be internal entity in the same time, in the case if they are copies (for example, copies to archive, saved data). The column, which does not contain neither “r” nor “u” and not outgoing external entity, must be deleted. There is no function using it, the system does not need it.
 7. In the case, a row contains more “c” and “h”, worth to try to decompose that function in sub-functions or processes. Every row contains 1 “c” or ‘ “h” at ideal case.
 8. The 3.-7. steps must be repeated until a cycle had run without modifying any rows or columns. That means that the matrix correspondences all the above conditions.

If the number of the “c”-s are bigger than 1 (rule No. 4), a general function may be imbedded: for instance both the ambulant intervention, both the surgical intervention create an intervention during sustenance. At this time, in the rows of the above mentioned two functions the “c” must be replaced with an “h” and it must be noted that the functions contain a newly created general function.

Function/entity	Intervention		Function/entity	Intervention
Carry out ambulant intervention	c	→	Carry out ambulant intervention	h
Carry out operation	c		Carry out operation	h
			A/insert intervention	c

During the operations of the c/u matrix, modification or correction on the list of entities or functions always happen, or modification or correction in their relations. That is why it is no

use defining the final form of the functional decomposition without the c/u matrix reaching its final form [9].

Diagonalising the c/u matrix

Diagonalising means the regrouping of the functions, during which the letters “c” are placed near the main diagonal of the matrix. Diagonalising is carried out with exchanging rows, then columns.

The relations in time between single processes are only meaningful within one function, as the functions themselves are without time-dimensions. It is necessary, though, to group the functions by their areas of application (scopes). This grouping is carried out by extant experience, so it is based on some knowledge of a certain level about the work processes of the organization.

The rows of the c/u matrix first have to be grouped by business areas (functional areas), then the elements of the groups created, have to be assigned to the functions within the group, then to the sub-functions within the functions (if there are sub-functions). Since these smaller aggregations are more easily transparent, the sequence of the rows (processes) within the functions is more easily allocated [32].

Exchanging rows is done continuously, closing the columns to the left by c-d-h-u..

The functions enumerated in the user’s requirements can be grouped by the following:

The functions providing system maintenance do not fit in the arrangement by the life-cycle, to these, a code which is bigger than the code of the last business area (main function) has to be assigned, while the general functions which also do not fit in the arrangement, have to have the biggest code assigned to them. By doing this, when arranged, these functions will be grouped, but will be under the sequence functions in the rows of the c/u matrix (see appendix).

After exchanging columns, the “c” letters of the specific functions will be above the main diagonal of the matrix.

After diagonalising has been complete, 127 functions (process) and 173 entities will remain in the system.

These are parallel, or analogue functions that are separated in the life of the organization but their processes are similar, and the entities used by them are the same. Illustrating the parallel functions in medical sustenance, or in the system of a hospital is of special importance. The modern quality control system expects the in- and out-patient sustenance, to handle the documentation of the patient in a standardised way, irrespective of

the form of sustenance. The documentation system of the in- and out patient cases is partly compatible, on the other hand there is a passage between the two systems since certain out-patient cases must be documented with in-patient documentation and vice versa owing to the data providing commitment by the financial system. In order to avoid two documents to be treated the same way during different procedures, general functions are imbedded.

Note, that even during the steps of diagonalising, modifications occur in enumerating functions and entities, and in their relations.

9. Results

Creating business areas

The sub-matrices of the diagonalised c/u matrix define the main functions of the information system of the hospital, and they also define the optimal structure of the provided services in the aspect of the system. This is made by regrouping the business areas, and within them, regrouping functions.

Appointing business areas is defined by the rules of business logic. These are:

- Maximal internal binding

Within the business area (that is, in the sub-matrix) there are few empty cells. The functions of the business area exchange data with each other or use the same entities. The data exchange, the technological order of the functions and using the same resources make grouping reasonable. The optimal grouping of the data base elements and the resources saves time and cost both in the course of working and operating the system.

- Minimal external binding

Outside the business area, there are few letters "u". The data exchange between the business areas must be reduced to the minimum. Its aim is not to reduce the communication of certain areas but to regroup data exchange in a way that it should primarily go within the activity areas. The minimal external binding will not reduce itself to zero, but this is not the aim, either.

- Logically created areas

The designers of the system must remember that it is not the organization of the procurer that is for them, but it is them who have to work on making the activity of the procurer more efficient and more effective with the tools of information technology. For that very reason, if the rules of business logic are not taken into consideration, then it can result in designing mathematically fine but otherwise useless systems.

The above-mentioned aspects effect against each other, therefore, in each case, the compromise of the common sense is necessary to have the solution. The preponderance of certain aspects leads to distorted systems [28]:

- The preponderance of the maximal internal binding
One row is a business area, defined by the letters “c” of certain rows. The number of empty rooms within business areas: 0. Each function has the life of its own; the company is made of incomprehended, solitary geniuses.
- The preponderance of minimal external binding
The whole sub-matrix – defined by rows of special functions – is a single business area. Not a single letter is left outside. All activities are poured in one small ball. The incompetent boss superbly disguises his being a dud.
- The preponderance of the traditional business logic
No function rearrangement is carried out; everything is “good as it is”. In this case the system results in the speeding up of the traditional data exchange in the life of the organization with the tools of information technology. Almost all companies that had no information technology past were ambushed by the spread of the IBM PCs in offices in the 1980’s. The anachronistic, low-efficiency organization did everything the same wrong way as before, but more efficiently.

There is a reason to appoint (partial) business areas within a business area or vice versa, to close certain business areas up. This is the description of the fine structure of the system. Giving names that express business areas makes both the organization and the system more perspicuous [45][33].

It should be borne in mind, that business areas must be amended with general functions, therefore, the new FD by using them. The fact, that the general functions are illustrated as the utilities of the system, irrespectively of the business areas, means that they can be developed or supplied independently, irrespectively of the other phases of the system-developing project.

Business areas describe activities, which take place in the organization; consequently, they are to be identified as a sub-system, or a system module of a well-defined organization unit, or (a) scope(s) of activities. When making the project of realisation of the system it is very useful to appoint them both in case of planning the cost and the timing: during the realization it is no use distributing into smaller units than the business areas, though, when defining the sequence of realisation, the following factors of the business area play an important role:

- How much time it takes to realise it.
- How much it costs.
- How problematic it is in the life of the organization.
- How urgent it is.
- Where it concerns the other activity areas.

Data flow diagrams

Data flow diagrams illustrate the data flow between functions. The transaction number between certain workplaces provided by certain functions represents the relations of the system elements and also helps to model the necessary bandwidths. The data flow diagram only contains the sequence functions.

The resolution of the data flow diagrams – similarly to the resolution of geographical maps – is marked with indices: by these, the data flow is illustrated by the DFD (0) (Dataflow Diagram 0) among the business areas (function groups, main functions), while the DFD (1) is illustrating the data flow within the business area in higher resolution (s.í.t.).

Illustrating data flow diagrams with the help of the diagonalised c/u matrix becomes immensely simple:

- The letter “r” that is near or outside of the business area or function stands for the incoming entity.
- The letter “r” that is under or above or outside of the business area or function stands for the outgoing entity.
- The “c”-s of the columns with no “r”-s mean outgoing entity.
- The “r” –s of the columns with no “c”-s mean incoming entity.

According to the above definitions all “r”-s mark data flow between 2 business areas or functions without containing the “r”.

It is apparent that implementing imbedded function does not ruin the possibility of mapping the c/u->DFD, all in all, the following modification has to be applied: in those columns where there is “h”, it has to be treated as “c” in the algorithm, and the “c” which is in the column must be ignored.

Data safety matrix

The function of data safety is to ensure the access rights to the users concerned, while constraining or denying unauthorised users’ access. Constraining the access [13]. Constraining access can cover

temporality (for instance, coming down to working hours)

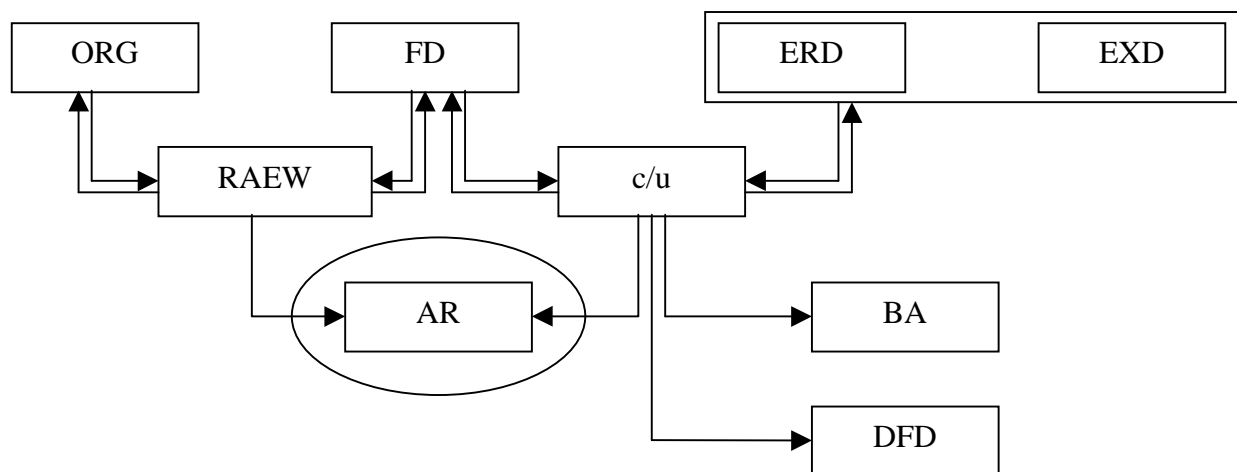
- localisation (defining terminals)

- operations (read, write, delete, run).

The organogram of the organization contains the positions doing activities. Certain access rights are assigned to certain positions (actors). The organogram illustrates the structural hierarchy, where all positions are only given once. Different people working in different positions do the activities of the organization; their relation is described in the RAEW matrix [27]. According to this, each person doing an activity, links to certain functions of the organization by his/her position (role) such as the role of a(n):

- R: responsibility
- A: authority
- E: expert
- W: work

The model based on the Martin-kind organizational model:

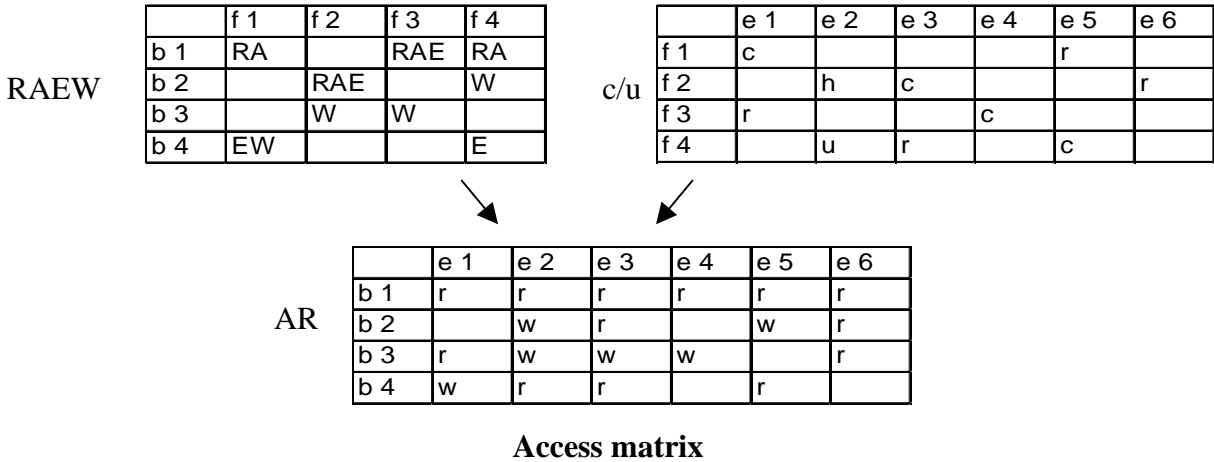


The enhanced organizational model

Marking:

- ORG (Organogram): organizational hierarchy
- FD (Functional decomposition): functional decomposition
- ERD (Entity Relation Diagram)
- EXD (External diagram): external relations
- RAEW (Responsibility, Authority, Expertity, Word)
- c/u (create-use)
- AR (Access Rights)
- BA (Business areas)
- DFD (Data flow diagram)

The AR-matrix contains the access rights that come into existence as the product of the RAEW matrix – which associates the functions of the organization and the organizational hierarchy – and the c/u matrix – which describes the relation of certain functions and their entities. By the RAEW matrix, to each function 4 position is linked, exactly an R, an A, an E and a W, and these functions handle an entity in 5 ways: c, h, r, u, d. According to this, a position handles an entity in 20+1 ways (the +1 means no treating at all). The model illustrated here handles this complicated system of relations automatically:
 The “w” of the RAEW authorises writing in the AR, while all other RAEW letter authorises reading in the relation with the c/u.



10. Summary

This PhD dissertation describes the way to determinate the system requirements for the hospital information system. It begins with the introduction of the hospital claims and goes on to describe the software specifications. It gives a brief summary on the evolution of the domestic hospital informatics, showing that disregarding of the requirements determination process result in a very low rate efficiency of this kind of systems.

The first part of the dissertation contains an overview on the roles and types of the requirement documentation, and the types of the requirements.

The second part of the dissertation describes step by step how to make the requirements documentation of an information system of a concrete hospital, and also contains the academic results of further implementing the planning method. After formulating the requirements of the hospital, the user’s requirements are defined. The user’s requirements

don't pan out about the non-functional requirements, partly because the procurer is not able to express its own demands, and partly because it takes the utility criteria for granted. The requirements of an area of specialty appear in connection with the secondary, non-healing-curing activities.

The steps of defining the system requirements begins with the definition of the task system of the institution, since the affectability of the system mainly depends on the data providing in connection with measurements expected from the system. The functional decomposition or functional analysis is detailing the functional requirements of the system in the decomposition of main function-function-process. Assigning them to the functions defined by the above-mentioned way makes defining the handled entities. That is, in case of each function or process it has to be defined what input parameters are necessary for the function to work and what outputs are produced the function or process in question.

The centre point of the method is the c/u matrix. This will be used to regulate the relation of the functions and the entities. Placing the functions of the system into the rows of the matrix, and the entities - handled by the system - into the columns, all possible function-entity relation are easy to review and define. During the operations of the matrix the system becomes complete and free from contradictions, then, while diagonalising, it defines the architecture and the data flow diagram of the system. The matrix operations have effects on the entities, as well; therefore, the final data model and the external relations can only be defined after the final form of the matrix has been achieved.

This method has not been widespread in the domestic routine because of two problems:

3. In the course of the diagonalising, two methods are known for defining the sequence of functions in the bibliography: the sequence by the life-cycle and the sequence by function-dependency. Sequence by the function-dependency is simple, can be used well in, for instance, production processes, but in the systems with feedback (like in health care) it is not a solution. Defining the sequence by the life cycle it causes problem that some functions cannot be fit in the sequence by the life cycle or could be fit in more places.
4. Functions and entities can have 4 kinds of relations in the c/u matrix. Marking these: „c”, that is, create, „u” that is, use (modify), „r” that is, read and „d” that is, delete. One entity can only have one “create” relation with a function; therefore, the functions have to be given exclusive rights to create entities. Solving this problem means difficulty in case of parallel or concurrent functions; for instance, a patient can get the same examination in the in-patient and out-patient specialist surgery, the same entities of casewith different forms,

in this case, the examination has to be recorded by both caseprotocol. Recording the examination results is carried out by strict protocol, defined by the sequence of the functions in both caseforms. Therefore, the function called “recording the examination results”, according to the life-cycle sequence, has to be stated in both the in-patient and the out-patient sustenance, which does not cause concern in itself, as renaming would make it evident which function is mentioned, for instance, “recording examination results in out-patient sustenance” and „”recording examination result in in-patient sustenance”. The problem is, that both functions would create the same entity, but in the c/u matrix in the row of an entity, only one “c” is allowed.

Solving the two problems together is not implementing general or sequence functions. The newly implemented general function contains the exclusive right to create entity. It is out of the sequences, but with invitation from the sequence function, the function of creating an entity gets imbedded into the calling procedure. The parallel functions can be handled free from contradictions, then. In case of the above mentioned example the newly implemented function “put in examination result” contains the “c” (create), while to the original two functions “h” (have create) function can be assigned, which shows that the function is question will contain an imbedded, general (non-sequence) function, and it has to mark this way in the functional decomposition, as well.

After introducing the new function-entity relation, the c/u matrix still can be mapped to the data flow diagram, so this function is not ruined. Mapping only needs a small change, namely, in those columns where there is an “h”, the rule for “c”-s has to be applied on them.

The dissertation contains the rules of creating business areas of the system and presents their application. It contains the functional decomposition of a complete system, with the sequence and the non-sequence functions, and a group of functions called system functions, which did not fit in either of the above-mentioned groups, as they are neither sequence, nor imbedded general functions. It illustrates the diagram that describes the external relations of the system, and the data model of the complete system.

The thesis presents how to make the data safety matrix of the complete system simply. It is an elemental part of the organizational model itself, therefore, any data providing requirements, functional change, or reforming the structural build-up automatically inherent the actualisation of the data safety matrix.

Finally, the dissertation contains the non-functional requirements of the information system of the hospital model.

11. Irodalomjegyzék / References

1. Bagchi, Ashutosh: A study on managing the performance requirements of a distributed service delivery software system, *Information and Software Technology* Volume 47, Issue 11 , 1 August 2005, Pages 735-746
2. Beyer, H.&Holtzblatt,K. (1998). *Contextual Design: Designing Customer-Centered Systems*. San Francisco, CA: Morgan Kaufmann.
3. Bodnár, Viktória: *Controlling - az alapoktól az aktualitásokig*. Budapesti Corvinus Egyetem, egyetemi jegyzet
4. Booch, G., Jacobson, I. and Rumbaugh, J.: *The Unified Model Language User Guide*, Addison-Wesley, 1999.
5. Bordás I. és munkatársai: *Fekvőbeteg intézmények endo-finanszírozási modellje és információrendszer egy új finanszírozási rendszerben*. Kórház, 1996
6. Calle, J.E., Saturno, P.J., Parra, P., Rodenas, J., Pérez, M.J., F. San Eustaquio and E. Aguinaga: *Quality of the information contained in the minimum basic data set: Results from an evaluation in eight hospitals* *European Journal of Epidemiology*, Issue: Volume 16, Number 11, November 2000, P:1073 - 1080 .
7. Chen, P. "The ER Model Toward a Unified View of Data", *ACM TODS* 1(1) 1976 pp. 9-36
8. Chen, Peter: *The entity relation model toward a unified view of data*, *ACM, Transaction on Database Systems*, 1, 9-36
9. Christensen T, Grimsmo A.: *Development of functional requirements for electronic health communication: preliminary results from the ELIN project*. *Informatics in Primary Care*, 2005;13(3):203-8.
10. Daragó, László: *Some aspects of teaching the technology of designing and planning information systems in health care*, *Teaching of Mathematics and Computer Science*, 3(2005)2 (131.-144.)
11. Daragó, László: *Szükség van-e egészségügyi informatikusra? XXII. Neumann Konferencia kiadványa*, 96-97.o., 2000. November 9-10, Veszprém.
12. Davis, Gordon B., Olson, Margarethe H.: *Management information systems: Conceptual Foundations, Structure and Development*, McGraw-Hill, 1987.
13. Davis, M.: *Software Requirements: Objects, Functions and States*. Prentice-Hall, 1993.
14. Dévényi, Dömötör: *Infokommunikációs és globális technológiák az egészségügyi informatikában*, IME I/1. 2002.

15. Dick, RS.; Steen, EB. The Computer-based Patient Record - An Essential Technology for Health Care, Revised Edition. Washington D.C., Institute of Medicine, National Academy Press; 1997.
16. Doll WJ, Torkzadeh G. The measurement of end-user computing satisfaction - theoretical and Methodological issues. *MIS Quarterly*. 1991;15:5–10.
17. European Committee for Standardization (CEN). European Prestandard - prENV 12265. Medical Informatics Electronic Healthcare Record Architecture., 1995.(HISA)
18. Gause, D. C. and Weinberg, G. M. :Exploring Requirements: Quality before Design. Dorset House, 1989.
19. Gervas J, Perez Fernandez M.:Minimum basic data set in general practice: definitions and coding. *Family practice.*, Oxford university press, 1992 Sep;9(3):349-52.
20. Gibson, Bill: Top-Down System Design,
http://msdn.microsoft.com/vstudio/teamsystem/reference/technotes/system_designer/topdown_sys.aspx
21. Grimshaw, David J., Draper, Godfrey W.: Non-functional requirements analysis: deficiencies in structured methods, *Information and Software Technology* Volume 43, Issue 11 , 1 October 2001, Pages 629-634
22. Huotari, Maija-Leena, Wilson, T.D.: Determining organizational information needs: the Critical Success Factors approach. *Information Research*, Vol. 6 No. 3, April, 2001
23. Kotonya, G. and Sommerville, I.: *Requirements Engineering: Processes and Techniques*. Wiley, 1998.
24. Kuhn K, Lenz R, Blaster R.: Building a hospital information system: design considerations based on the result from Europe-wide vendor selection process.In: Lorenzi NM (ed). *Proc. AMIA Symp 1999*;:834-838.
25. Laerum H, Karlsen TH, Faxvaag A. Impacts of scanning and eliminating paper-based medical records on hospital physicians' clinical work practice. *J Am Med Inform Assoc*. 2003;10:588–595. doi: 10.1197/jamia.M1337.
26. Leisind, M.:Gyógyszerelosztás – hiányos minőségbiztosítás, *Management & Krankenhaus*, 2002.
27. Lewis, William J.: How Do You Spell Data Warehousing Success?, *DMReview*, April 15, 1999 Issue
28. Lor, Kar-Wing Edward: Operational Definitions for System Requirements as the Basis of Design Automation, *Software - Practice and Experience*, Vol. 21(10), 1103-1124.
29. Martin, J. McClure, C.:*Structured Techniques, The Basics for CASE*, Englewood Cliffs, NJ:Prentice Hall, 1988.

30. Martin, James, Cleve Finkelstein, Information Engineering. Savant Institute, England, 1981.
31. Martin, James, Joe Leben: Strategic information planning methodologies. Prentice-Hall Inc., 1987.
32. May, R. M.: Simple mathematical models with very complicated dynamics, Nature 261. kötet, 459–467. o. (1976)
33. Moriarty, T., Thompson, V.: Business Analysis Techniques, Database programming and design, aug. 1996.
34. NM rendelet, 9/1993. (IV. 2.), Az egészségügyi szakellátás társadalombiztosítási finanszírozásának egyes kérdéseiről
35. O'Brian, James A.: Introduction to information systems, Richard D. Irwin Inc., 1991.
36. O'Brian, James A.: Introduction to information systems in business management, Richard D. Irwin Inc., 1991.
37. Pedersen, T. B. and Jensen, C. S.. Research Issues in Clinical Data Warehousing. In Proceedings of the SSDBM Conf., pages 43--52, July 1998.
38. Politiano, A. L.: Salvaging Information Engineering Techniques in the Data Warehouse Environment. Information Science, Vol. 4. No.2, 2001. pp. 35-43.
39. Poollet, Michelle A.: Solutions by Design: The Security Matrix, SQL Server Magazine, March 2000
40. Raffai, Mária: UML Modellező nyelv, RUP Módszertan, Novadat, 2002(UML)
41. Snooke, N. A. and C. J. Price (1998). Hierarchical functional reasoning. Knowledge-Based Systems 11 (5–6), 301–309.(FD)
42. Sommerville, I. (2005). 'Integrated Requirements Engineering'. IEEE Software, 22 (1), 16-23, January/February 2005.
43. Sommerville, I. 2002. 'Software Documentation'. In Software Engineering, Vol 2: The Supporting Processes. R. H. Thayer and M. I. Christensen (eds), Wiley-IEEE Press.[Book chapter]
44. Sommerville, Ian: Software engineering. Addison Wesley Pub Co; 6th edition, 2000.
45. Stamper, R.K.:Organisational semiotics. Informatics without the computer, in Liu K, Clare RJ, Andrsen PB, Stamper RK (eds. 2001). Information, organisation and technology. Studies in organisational semiotics. Kluwer Academic Press, Boston., pp. 115-171.
46. Thayer, R. H. and Dorfman, M.: Software Requirements Engineering. IEEE Computer Society Press,1997.

47. Török Kovács, Anett: Az egészségügy-finanszírozás új rendszere Magyarországon, Polvax, 1997/2.(1. évf. 2. Sz.)
48. Trevino, FM. Uniform minimum data sets: in search of demographic comparability. *American Journal of Public Health* 1988;78:126–7.
49. Uri Katalin: Globális technológiák az egészségügyi informatikában, IME II/5. 2003.
50. Viller S. and Sommerville I. 2000. 'Ethnographically informed analysis for software engineers'. *International Journal of Human-Computer Studies*, 53 (1) : 169-196.
51. Walsham G: Interpretative case studies in IS research: nature and method. *European Journal of Information Systems*, No. 4, Operational Research Society, Great Britain, pp. 74-81.
52. Waters, Donald J., "We have a Computer: Administrative Issues in the Relations Between Libraries and Campus Computing Organizations." *Journal of Library Administration* 13, no. 1 (1990): 121.
53. Whitten, J.L., Bentley, L.D. and Dittman, K.C. (2001) 5 th ed., *Systems Analysis and Design Methods*, Irwin/McGraw-Hill, New York, NY. Chapters 5,8. Hoffer, J.A., George, J.F. and Valacich (1999) 2nd ed., *Modern Systems Analysis and design*, Benjamin/Cummings, Massachusetts.
54. Widom, J.. Research Problems in Data Warehousing. In *Proceedings of CIKM 1995*, pp. 25–30.
55. Woodsworth, Anne: *Patterns and Options for Managing Information Technology on Campus* (Chicago, American Library Association, 1991): 74-75.

12. Publikációs lista / List of publications

Idegen nyelvű referált cikk:

1. László Daragó, Pierry Lévy: Caseview_hun: un outil internet pour appréhender les DRG hongrois, Journal d'Économie Médicale, 2003. Vol 21. n° 7-8, 451-454.
2. László Daragó: Some aspects of teaching the technology of designing and planning information systems in health care, Teaching of Mathematics and Computer Science, 3(2005)2 (131.-144.).

Magyar nyelvű referált cikk:

3. Daragó László: Javaslat a HBCs rendszer korrekciójára, Egészségügyi gazdasági szemle, 41/3. 2003 május. 35-38.o.
4. Daragó László: Caseview_hun: eszköz a homogén betegségcsoportok (HBCs) áttekintéséhez, Egészségügyi gazdasági szemle, 42/1. 2004 január.47-50. o.

Magyar nyelvű szakkönyv:

5. Általános egészségügyi ügyvitelszervezői ismeretek, Medicina könyvkiadó Rt., Budapest, 2000. (két fejezet,159-171,227-241.o.), ISBN 963 242 620 7

Konferencia kiadványban megjelent cikk (külföldi):

6. Daragó László: Improving the DRG system in Hungary, The new navigators: from Professionals to Patients. Proceedings of MIE2003, St. Malo, France, IOS Press Nieuwe Hemweg 6b 1013 BG Amsterdam, 818-823., 2003.
7. László Daragó: Caseview_HUN: easy DRG overview, Transformation of Health care with Information Technologies, IOS Press Nieuwe Hemweg 6b 1013 BG Amsterdam, Th Netherlands, 182-189.,2004.
8. Pierre Lévy, László Daragó, Claire Beguin, Francis H. Roger France: application of the Caseview method to the Hungarian, Belgian and French DRG classification, PCS/E 20th International Working Conference, 2004. Október 27.-30. Konferencia kiadvány (CD, 10 oldal)
9. Pierre P. Levy, Laetitia Duché, Laszlo Darago, Yves Dorleans, Laurent Toubiana, Jean-Francois Vibert, Antoine Flahault: ICPCview: Visualizing the International Classification of Primary Care, Connecting Medical Informatics and Bio-Informatics, Proceedings of

MIE2005, Geneva, Switzerland, , IOS Press Nieuwe Hemweg 6b 1013 BG Amsterdam , 623-628., 2005.

Konferencia kiadványban megjelent cikk (hazai):

10. Daragó László: A megfelelő információ, A Kórházi Informatika és Menedzsment Aktuális Kérdései I. konferencia, 55-63.o., Nyíregyháza, 1996
11. Molnár K., Szegedi J., Valenta B., Fehér M., Daragó László: A diabetes gondozás minőségfejlesztésének informatikai módszere, MBT, 1996. október 11-12. konferencia anyag (4 oldal)
12. Daragó László, Szegedi J., Görögh S.: Az informatika jelentősége a szűrési, gondozási, megelőzési tevékenységben, XX. Neumann konferencia anyaga (220-222.o.), 1996. november 14-16.
13. Komoróczy Tamás, Daragó László: A Jósa András Megyei Kórház I. Belgyógyászati információs rendszerének kiépítése, XX. Neumann konferencia anyaga (157-159.o.) oldal), 1996. november 14-16.
14. Daragó László: Szükség van-e egészségügyi informatikusra? XXII. Neumann Konferencia anyaga, 96-97. o., Veszprém, 2000. November 9-10.
15. Daragó László: A művese állomás adatcseréje külső alkalmazásokkal, Informatika a felsőoktatásban 2002, Debrecen, 902-906.o.,2002. Augusztus 28-30.
16. Daragó László: Elosztott erőforrású rendszerek, MTA Sz-Sz-B.- Megyei Tudományos Testülete 10 éves Jubileumi közgyűléssel egybekötött Tudományos Ülésének előadásai, 269-274.o., Nyíregyháza, 2002. Szeptember 29.
17. László Daragó: A polynomial interpolation, ICAI 6th, Eger, 2004., konferencia kiadvány (315-319. o.)

Egyéb cikkek:

18. Daragó László: Önellátás helyett. Computer Panoráma, 99/7. Szám, 24-26.o.
19. Dr. Szegedi János - Daragó László: Az osztályos vezetői rendszer koncepciója, Kórház, 97/5, 43-47.o.

Konferencia előadások

1. Önellátás helyett. A Kórházi Informatika és Menedzsment Aktuális Kérdései I., Nyíregyháza, 1996. Január 25-26.
2. Információs rendszerek tervezése és megvalósítása
Országos konferencia az MPKITK szervezésében, felkért előadóként
Nyíregyháza, 1999. Május 25.
3. Szükség van-e egészségügyi informatikusra?
XXII. Neumann Konferencia, Veszprém, 2000. November 9-10.
4. Menopausa szakrendelés ügyviteli rendje.
Magyar Menopausa Társaság V. konferenciája Balatonfüred, 2001. Június 7.-9.
5. Daragó László: A művese állomás adatcseréje külső alkalmazásokkal
Informatika a felsőoktatásban 2002, Debrecen, 2002. Augusztus 28-30.
6. Daragó László: Elosztott erőforrású rendszerek
MTA Sz-Sz-B.- Megyei Tudományos Testülete 10 éves Jubileumi közgyűléssel
egybekötött Tudományos Ülése, Nyíregyháza, 2002. Szeptember 29.
7. Daragó László: Ki fizesse a kórházak rezsijét?
Magyar Tudomány Napja, Nyíregyháza, 2002. November 11.
8. Daragó László: Javaslat a HBCs-rendszer javítására, DE EFK, 2002. December 9.
9. Daragó László: Improving the DRG in Hungary, MIE2003, St. Malo, 2003. Május 3-8.
10. Rácz Krisztina, Daragó László*, Pap Károly, Kornafeld János
(Jósa András Megyei Kórház, Nyíregyháza, DE OEC Egészségügyi Főiskolai Kar,
Nyíregyháza*): Változókorai panaszok epidemiológiai értékelése tízezer perimenopausális
adatai alapján.
Magyar Menopausa Társaság V. konferenciája Balatonfüred, 2003. Június 13.-15.
11. Daragó László: Improving the DRG in Hungary, MIE2003, St. Malo, 2003. Május 3-8.
12. László Daragó: A polynomial interpolation, ICAI 6th, Eger, 2004. Január 27.-31.
13. László Daragó: Caseview_hun: easy DRG overview, PRO-ACCESS: 2nd International
conference in e-health in common Europe, Krakow, 2004. március 11.-12.
14. Pierre Lévy, László Daragó, Claire Beguin, Francis H. Roger France: application of the
Caseview method to the Hungarian, Belgium and French DRG classification, PCS/E 20th
International Working Conference, Budapest, 2004. Október 27.-30.
15. Pierre P. Levy, Laetitia Duché, Laszlo Darago, Yves Dorleans, Laurent Toubiana, Jean-
Francois Vibert, Antoine Flahault: ICPCview: Visualizing the International Classification
of Primary Care, MIE2005, Geneva, 2005. Augusztus 26-szeptember 1.

16. László Daragó, Pierre P. Lévy, Anett Veres, Zsolt Kristóf: ICDView: a tool to make the morbidity and mortality visible, View2006, Paris, 2006. Április 24.-25

Egyéb tudományos tevékenység:

- a VIEW2006 (Visual Information Expert Workshop) nemzetközi konferencia társelnöke; View2006, Paris, 2006. Április 24.-25. (www.view2006.org).
- Az Elsevier kiadó, reviewer a Health Policy folyóiratban megjelenő tudományos közlemény bírálatához. (<http://www.elsevier.com/locate/healthpol>).

Daragó László által fejlesztett és telepített egészségügyi szoftverek

Szoftver neve	Környezet/nyelv	Év	Megrendelő/felhasználó	Forrás (kb)	Dokumentáció
1. Infarktus regiszter 1.0	Turbo-Pascal/DOS	1987	Jósa András Kórház	2000 sor	
2. Infarktus regiszter 2.0	Clipper/DOS	1990	Jósa András Kórház	2000 sor	
3. Hypertonia regiszter 1.0	Turbo-Pascal/DOS	1987	Jósa András Kórház	1500 sor	
4. Hypertonia regiszter 2.0	Clipper/DOS	1993	Jósa András Kórház	1500 sor	
5. Hypertonia regiszter 3.0	Access/Windows	1998	Astra / Jósa András Kórház	4 Mbyte	Off line
6. Angiológiai regiszter	Turbo-Pascal/DOS	1987	Jósa András Kórház	1600 sor	
7. Diabetes regiszter	Turbo-Pascal/DOS	1987	Jósa András Kórház	1500 sor	
8. Elmeógyászati regiszter és statisztikai programcsomag	Turbo-Pascal/DOS	1988	Jósa András Kórház	1600 sor	Off line
9. Diabetesez Nephropathia regiszter	Clipper/DOS	1994	Kisvárdai Kórház	1600 sor	Off line
10. Menopausa szakrendelés 1.0	Turbo-Pascal/DOS	1987	Jósa András Kórház	3000 sor	
11. Qinput 1.0; OCR alkalmazás	Turbo-Pascal, Assembly	1990	Kiállítva: CeBIT '90, Hannover	1200 sor	
12. Menopausa Szakrendelés 2.0	Clipper/DOS	1993	Jósa András Kórház	3000 sor	
13. Menopausa Szakrendelés 3.0	Visual Foxpro	1998	Ciba Geigy / Magyar Menopausa Társaság	3 Mbyte	
14. Menopausa Szakrendelés 4.1	Access/Windows	1999	Magyar Menopausa Társaság	6 Mbyte	On line
15. Menopausa Szakrendelés 5.1	Access/Windows	2001	Magyar Menopausa Társaság	7 Mbyte	On line
16. Intézeti betegadminisztráció (IBE, IBEX, IBESZERV)	Clipper/DOS	1993	Jósa András Kórház, Kisvárdai Kórház, Pszichiátriai Szakkórház	10000 sor	Off line
17. Hemofilia regiszter	Access/Windows	2000	SMS / Heim Pál Gyermekkórház	5 Mbyte	Off line
18. Nephrológia szakrendelés és	Access/Windows	1999	EuroCare Magyarország	5 Mbyte	

Művese Állomás alkalmazás 1.0					RT.				
19. Nephrológia szakrendelés és Művese Állomás alkalmazás 2.0	Access/Windows	2000	EuroCare Magyarország	9 Mbyte	RT.	On line, Kontextus érzékeny help			
20. Nephrológia szakrendelés és Művese Állomás alkalmazás 3.2	Access/Windows	2002	EuroCare Magyarország	9 Mbyte	RT.	On line, Kontextus érzékeny help			
21. Nephrológia szakrendelés és Művese Állomás alkalmazás 3.6	Access/Windows	2003	EuroCare Magyarország	12 Mbyte	RT.	On line, Kontextus érzékeny help			
22. Nephrológia szakrendelés és Művese Állomás alkalmazás 3.7	Access/Windows	2004	EuroCare Magyarország	14 Mbyte	RT.	On line, Kontextus érzékeny help			
23. Nephrológia szakrendelés és Művese Állomás alkalmazás 3.8	Access/Windows	2005	EuroCare Magyarország	14 Mbyte	RT.	On line, Kontextus érzékeny help			
24. EuroCare bér-munkaügyi alrendszer	Access/Windows	2002	EuroCare Magyarország	4 Mbyte	RT.	On line			
25. Művese állomás jelentés készítés	Visual Basic 6	2004	EuroCare Magyarország	1 Mbyte	RT.	Magyar és angol nyelvű. Eddig kb. 7000 látogató			
26. Művese hálózat teljesítmény elszámolás	Visual Basic 6	2004	EuroCare Magyarország	1 Mbyte	RT.				
27. Caseview_HUN	Html, PHP, Mysql	2003	Freeware, elérhető: http://www.medinfo.hu/darago/caseview.php http://mail.de-efk.hu/~darago/caseview/caseview.php	1 Mbyte					
28. ICDView	Html, PHP, Mysql	2006	Freeware, elérhető: http://www.medinfo.hu/	1 Mbyte		On line Magyar és			

[darago/icdview.php](#)
<http://mail.de-efk.hu/~darago/icdview/caseview.php>

angol nyelvű.
Eddig kb.
7000 látogató