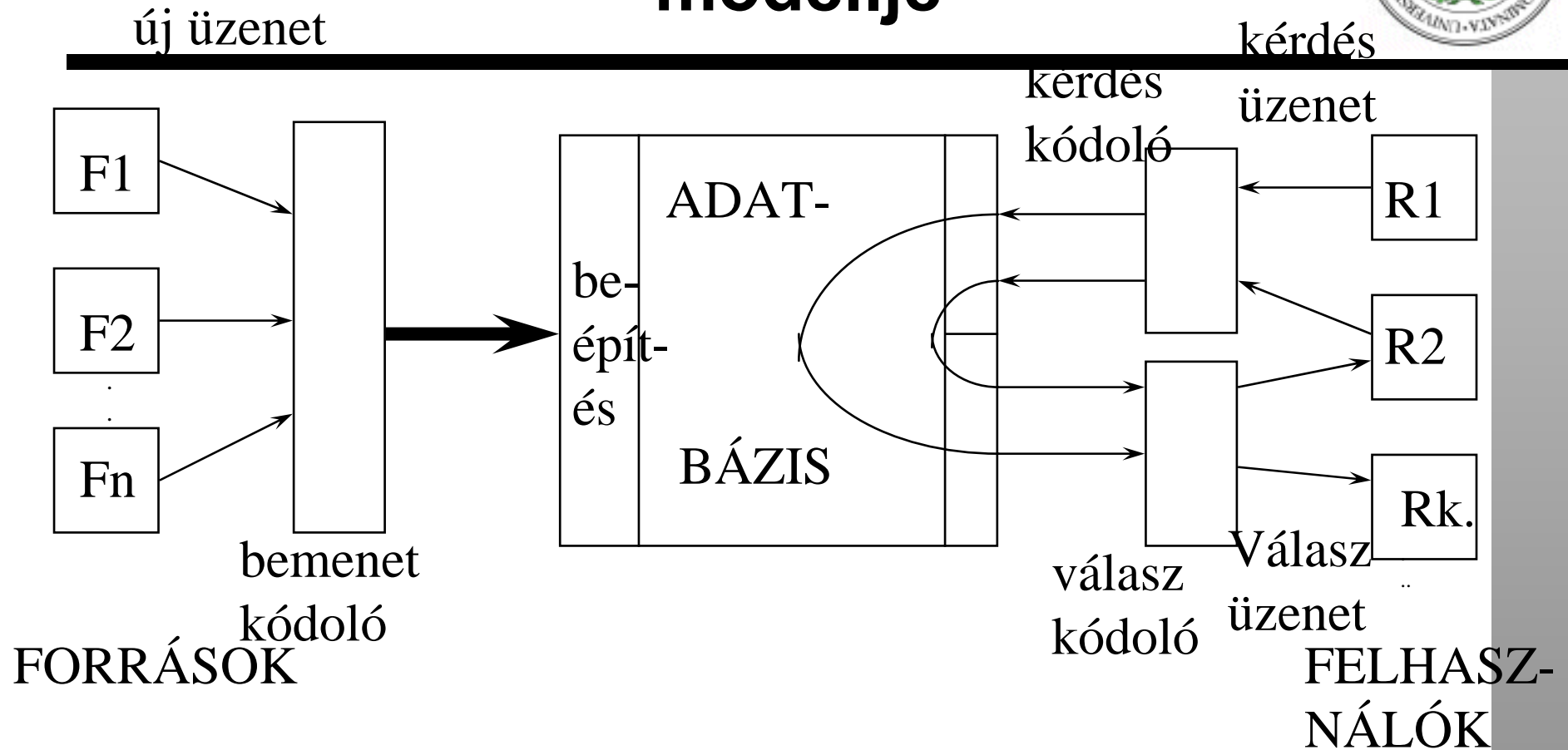




Információs rendszerek elméleti alapjai

Információelmélet

Az automatizált információs rendszer modellje



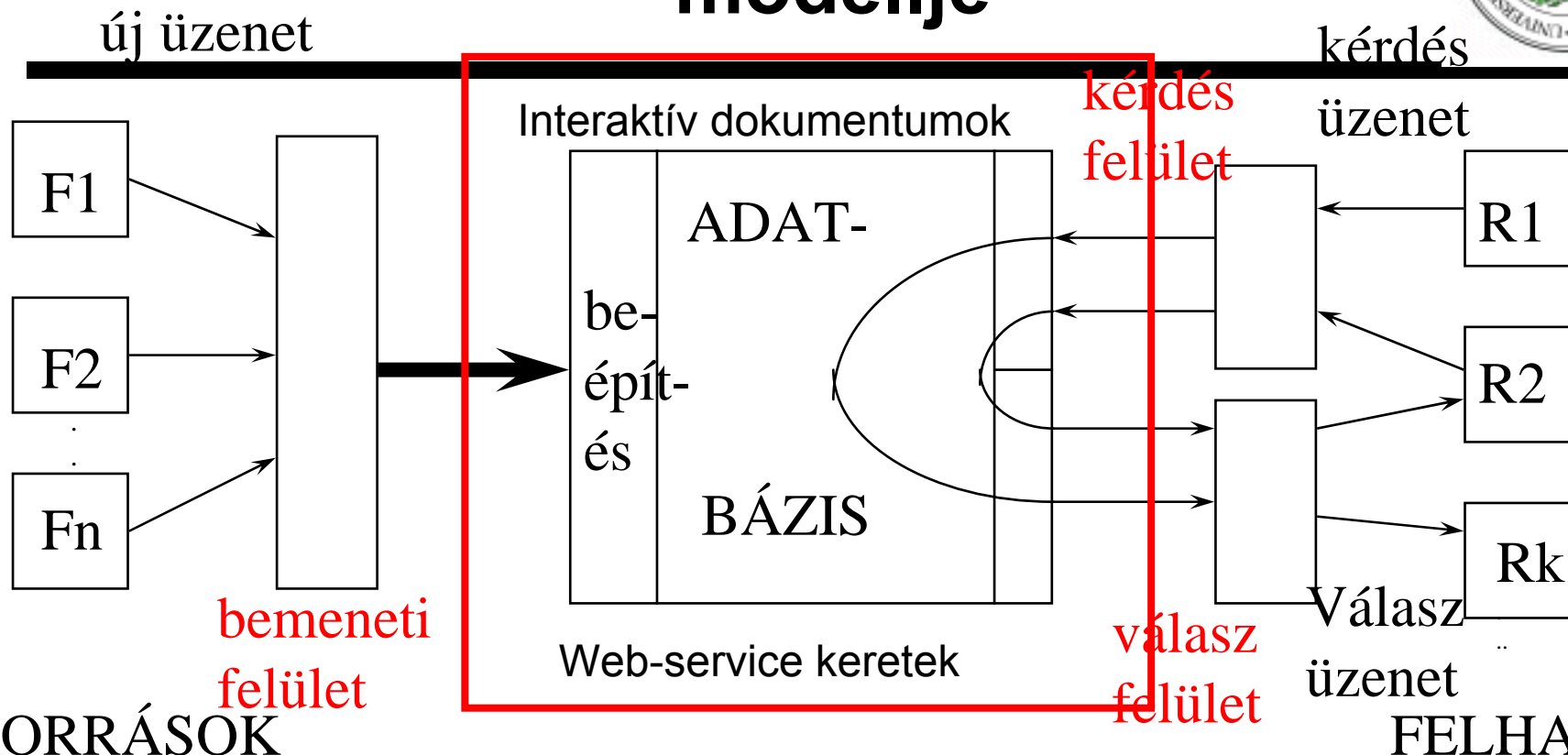
A kódolók előre rögzített formájú, speciális nyelven, szabvány szerint írt üzeneteket fogadnak, ehhez intelligens interaktív felületet adnak a források és felhasználók számára.



Ez a 20 évvel ezelőtti modell (Benczúr andrás), „közvetlenül” adatbázis-kezelőre építve.

Az interakciós felület mind az új információ bevitelére, mind az információ kinyerésére ma is programozott felület. A web-technológia új jellemzője, hogy az adatbázisba épülés, és az adatbázisból való információ-visszanyerés közvetlen folyamata és a felhasználói interakciós felület közé egy dokumentum (web-dokumentum) kezelése épül. Ezen a dokumentumon keresztül történik a felhasználóval való interakció és az adatbázissal való interakció. A régi sémára ezt a piros kiegészítéssel próbálom rárajzolni.

Az automatizált információs rendszer modellje



FORRÁSOK

A három **felület** egybeszervezett interakciós, „web-szerkeszthető” felületet jelent, speciális nyelven, szabványok szerint összeállítható dokumentumokat kezelnek a felhasználók. Az adatbázis-kapcsolatokat a Web-service rendszer automatizáltan kezeli.



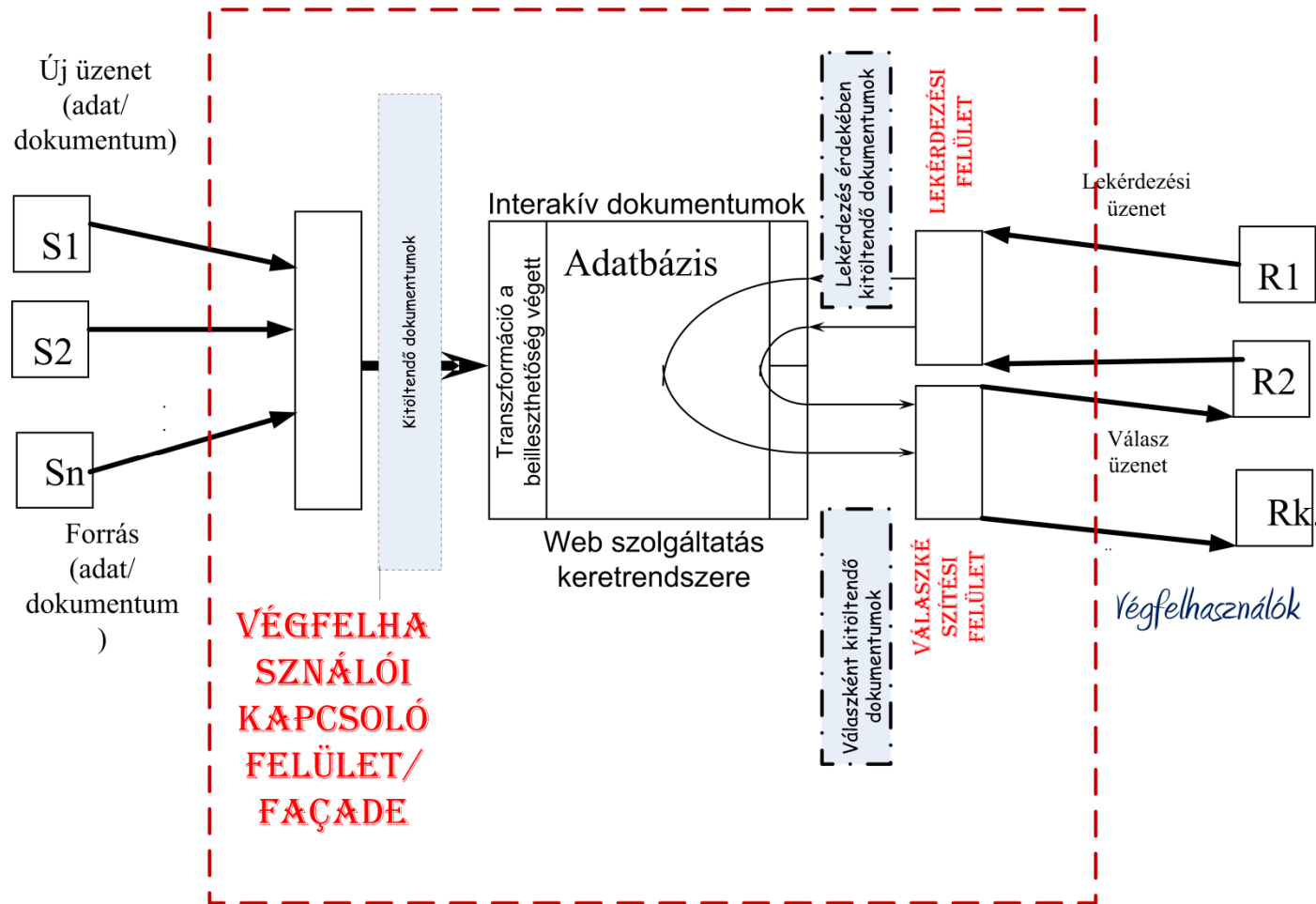
A rendszer programozása a személyi interakciós felületre és az adatbázis-interakcióra egyaránt szükséges.

Az adatbázis-tranzakciók dokumentumokhoz kötődnek, akár több mélységű hierarchiában. A tranzakciók korrekt kezelése az adatbázis-kezelő rendszerre tartozik.

A felhasználói felület dokumentumai félig-struktúrált adatok, kiegészítve prezentációs formával. Szerkesztés alatt szabad dokumentumok, ami azt jelenti, hogy változók is szerepelhetnek benne. A szerkesztés során a változók vagy a felhasználótól kapnak értéket, vagy az adatbázisból. A szerkesztés végén érvényesített dokumentummá, dokumentum részzé válnak, és a web-szerviz rendszere gondoskodik a megfelelő tranzakció végrehajtásáról. A dokumentumok korrekt kezelése a web-szerviz rendszerre tartozik. A dokumentumok szemantikus korrektsége, önmagukban és egymás között a szervezeti tevékenységekre, szervezeti egységek működésére vonatkozik.

Példaként a Bolkdijk-Blokdijk könyvből vegyünk egy kis rendszert.

Adatbázis-kezelés elvi modell





Példa

Nagykereskedési vállalat kereskedelmi részlege. Megrendelt **termékeket** szállít **vevőknek**.

1. A **megrendeléseket** ellenőrizni kell a vevő információival. Ha a rendelés korrekt, az információi kiegészülnek a vevő információival, (név, cím,) a további feldolgozáshoz.

2. A kiszállítandó termékek mennyiségei levonódnak a nyilvántartott raktárkészletből, a fizetendő összeg kiszámításra kerül és elkészül a **számla**. A számla teljes összege a vevő új **adóssága**.

3. A vevő **fizetése**i adósságai kiegyenlítésére beérkeznek.

4. Ha a vevő nem fizet egy adott időpontig (számla fizetési határidő), a vállalat **figyelmeztetést** küld a vevőnek.

A rendszer adminisztratív feladataival foglalkozunk, nem a fizikai termék, megrendelés fogadás, szállítás rendszerével.

ABKR: formális modell



Logikai modell

Valóság modellezése

Kórendszer: bináris szavak rendszere

Absztrakt adattípusok

Kiszámíthatósági bonyolultság elméleti elemzések

Adatmodellek

Előfordulások → kórendszer

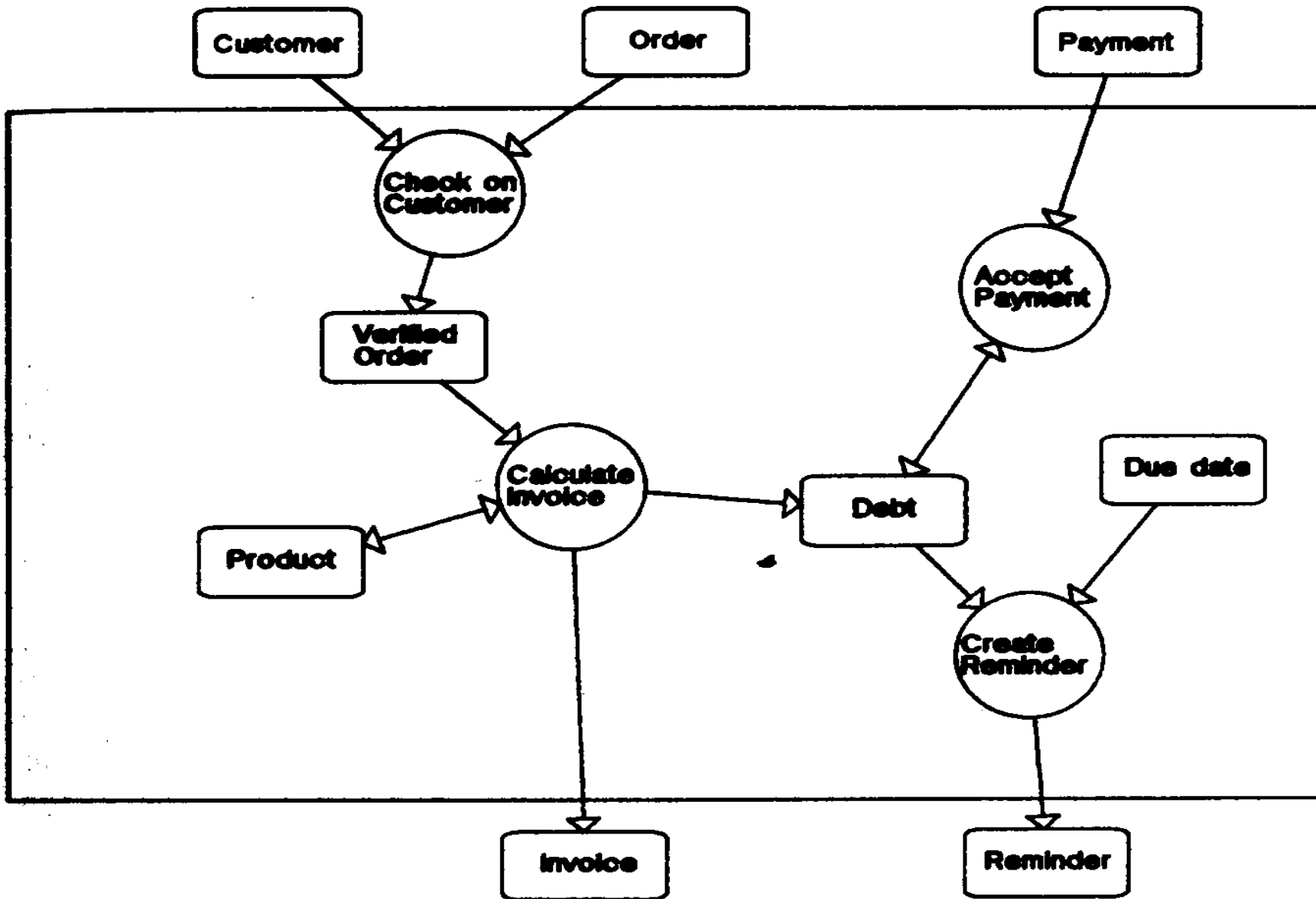
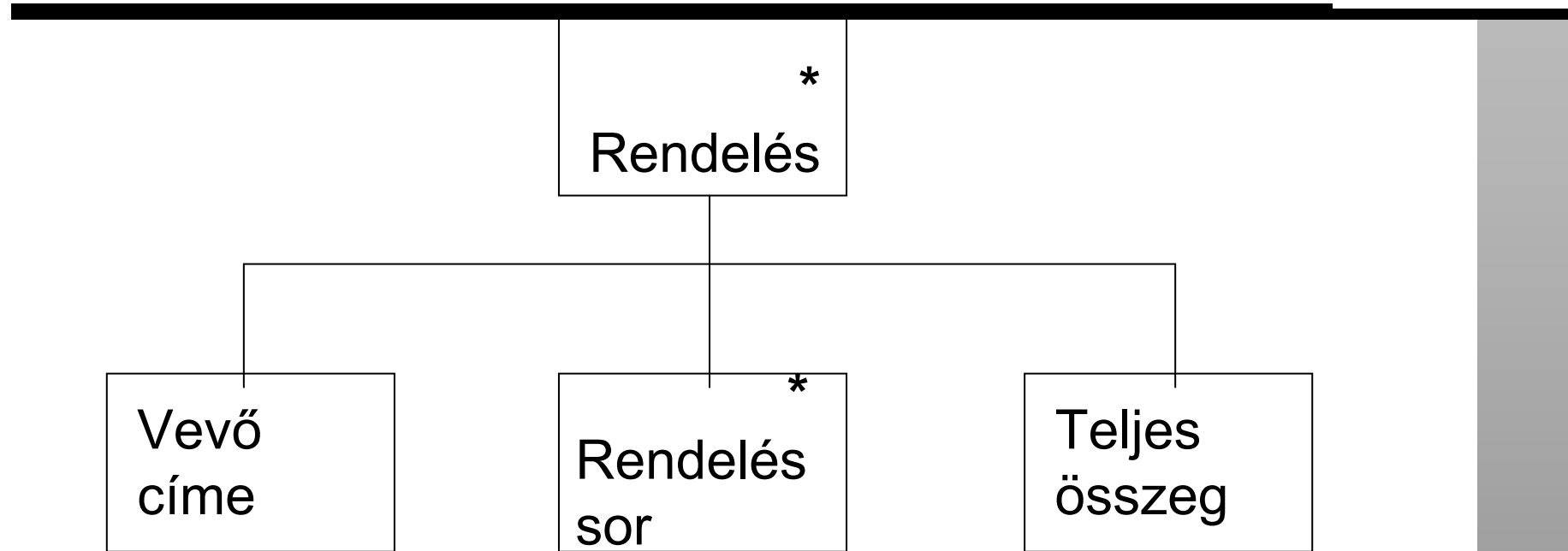


Fig. 47. Model of the Example

Számla előállítás végigkövetése



Jelölés:

* = többszörös részüzenet

O = feltételes részüzenet

= egyszerű részüzenet



Ez a felbontás még nem rendszer-dekomponálás.

A 65. Ábra rendszer-dekomponálással keresi a megelőző üzenetet, amelyből a részüzenet keletkezett.

Felhasználói feladatból indulunk, addig folytatjuk, amíg a rendszer inputjában meg nem találjuk a megfelelő részüzenetet.

A forrást keressük, ami lehet input üzenet, számított vagy származtatott üzenet más üzenetekből.

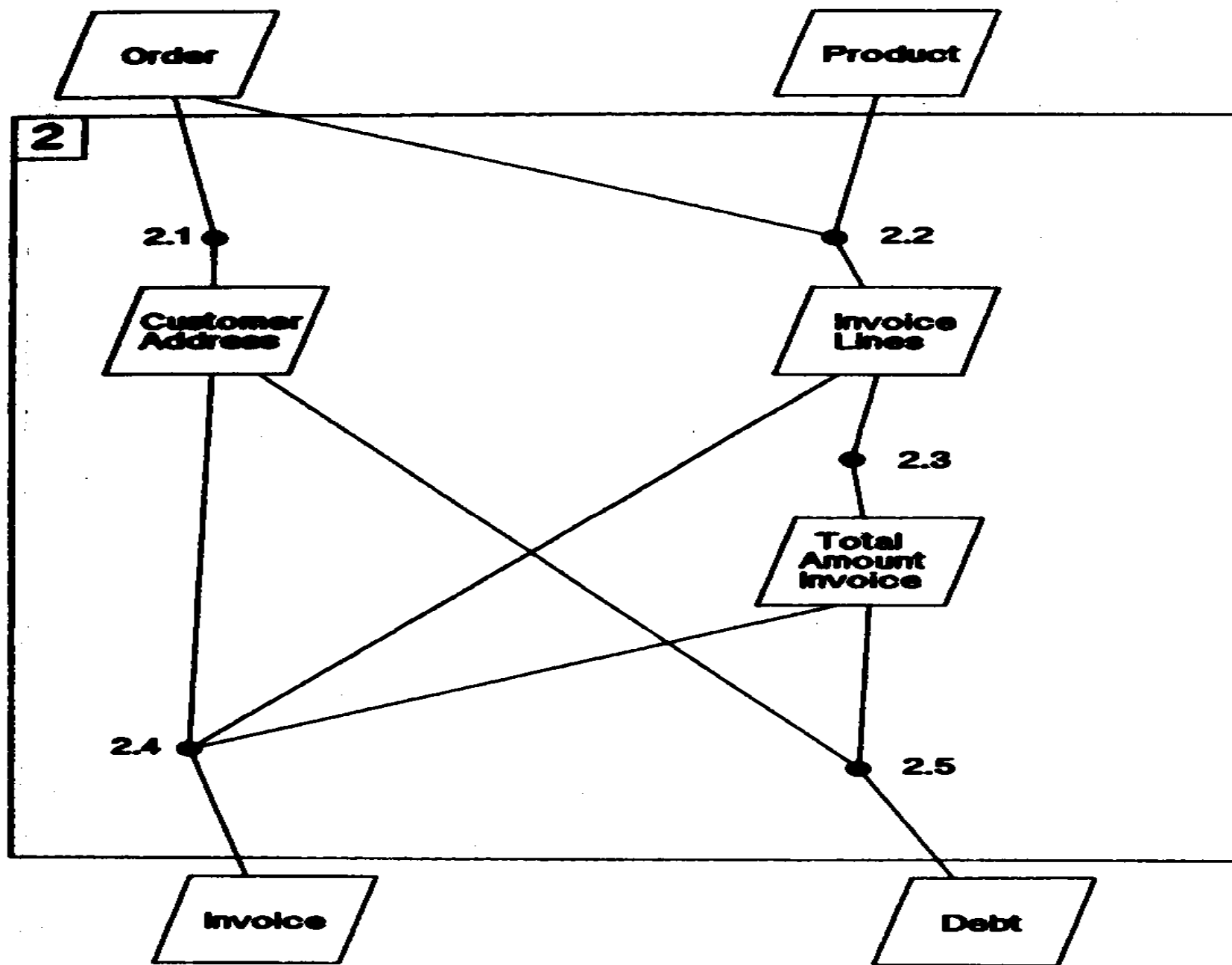
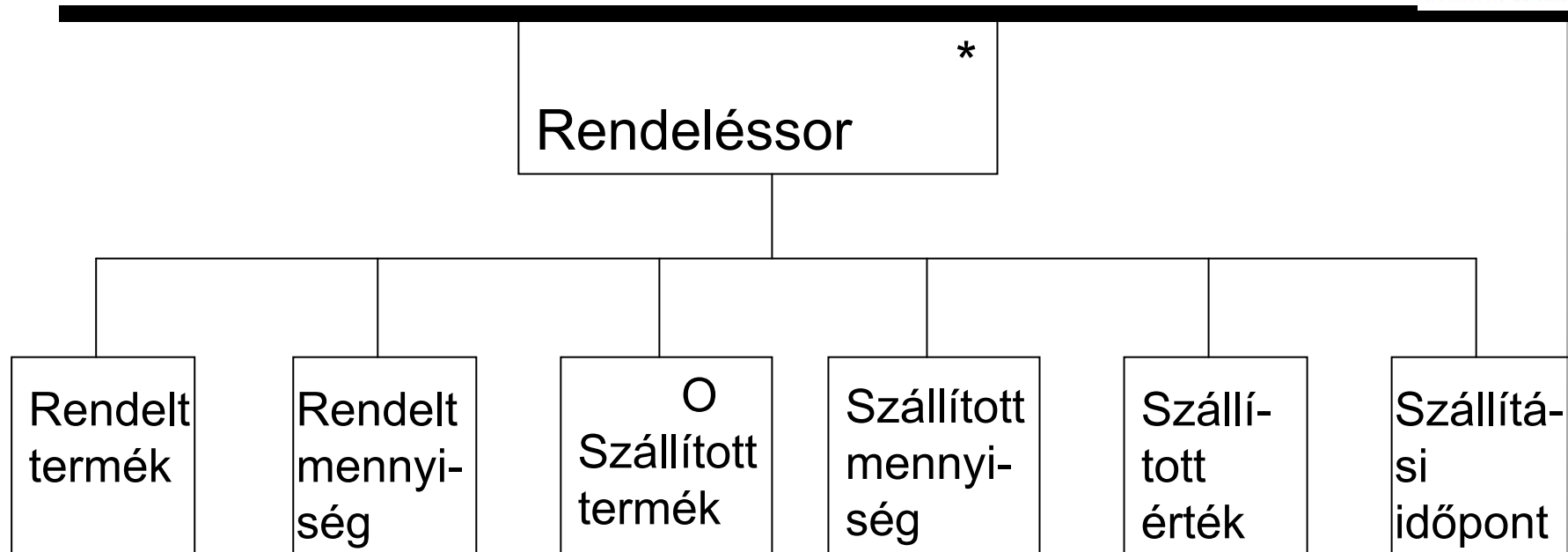


Fig. 65. Precedence Analysis for Submessages



A rendeléssor felbontása:



A 67. Ábra mutatja: minden elemi üzenet szintig le lett bontva. Elértük a belső határt.

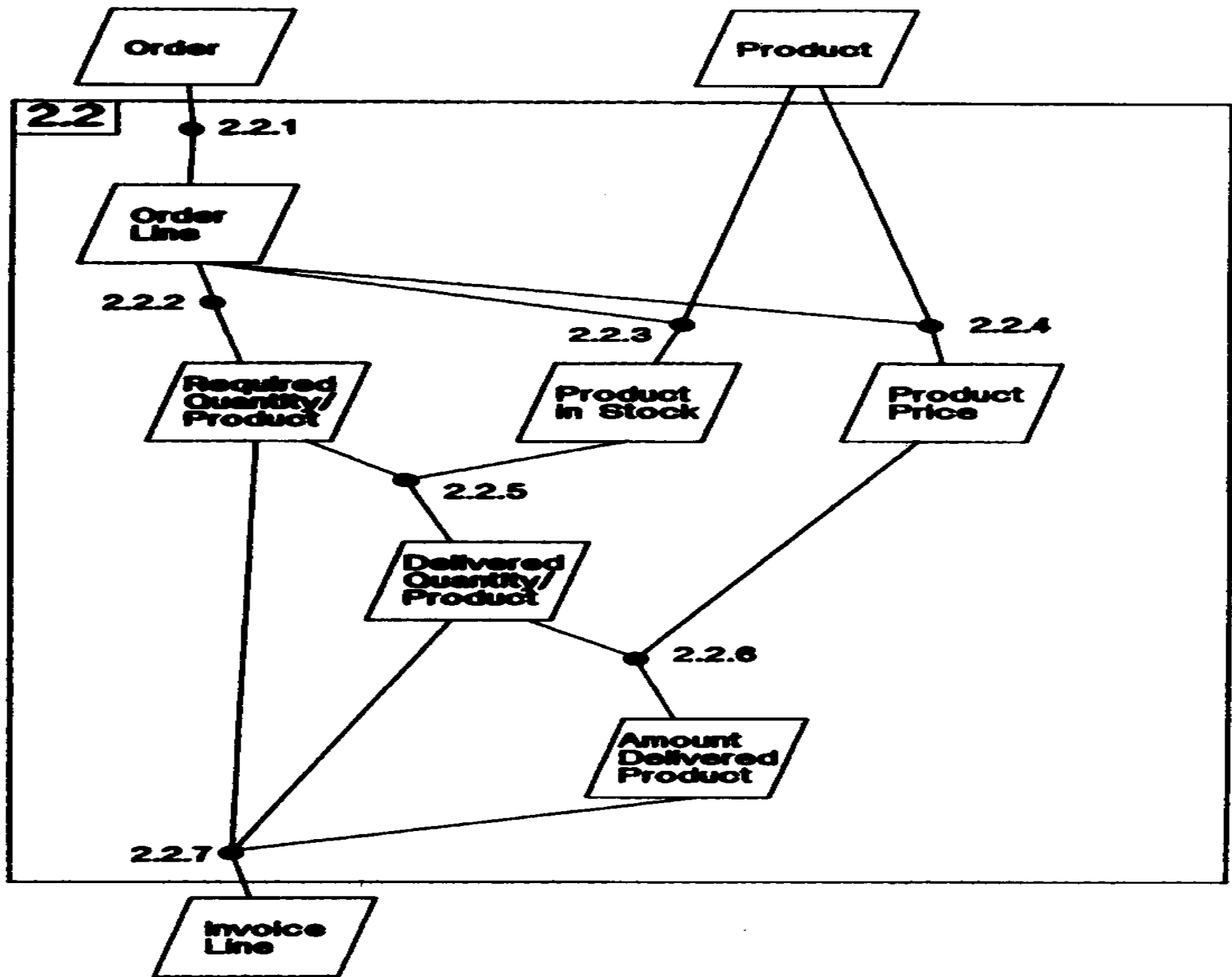


Fig. 67. Precedence Analysis of Subsystem: 'Make Invoice line'



7.8 Az eljárásmodell dekomponálása

Az információ dekomponálása impliciten kijelöli a részrendszereket. Fekete dobozok, belső tevékenységekkel. Elemi üzenet szintre lebontva már nem maradnak belső entitások, csak funkciók.

Funkcionális dekomponálás van hátra.

7.8.1 Elemi eljárások dekomponálása

Elemi részrendszer = elemi eljárás

Elemi lépésekből áll:

- Olvasás.
- Kiválasztás, gyakran bonyolult döntési tábla szerint.
- Számítás, különféle szelekciós tevékenységekkel és feltételekkel
- Tárolás, az információ megőrzése más felhasználói feladatokhoz.
- Szelektív output, üzenetek előállítása különféle feltételek szerint.
- Nyomtatás, írás, megjelenítés.

A 89-es (Benczúr A.) modell kibővítése



Újabb modell iktatódik a bemenetek és kimenetek közé.
Nem csak a módosítások és lekérdezések logikai leírása kell, hanem dokumentum-egységekbe szervezett interakciós szekvenciák, sőt újabban kollektívan kezelt dokumentumrendszerek illesztődnek az adatbázis elé.

A fizikai modell szintjén ez azt jelenti, hogy van egy speciális címtér, a dokumentumok címtere, innen kerül át az új aktivitást kiváltó kód az adatbázis fizikai címtérébe.

Egy dokumentum nem más, mint módosító adat, kérdés, válasz, módosító adat ,..sorozata, esetleg struktúrája.

Egy dokumentum-kitöltés független lépése addig tart, amíg nem szükséges másik forrástól további adat.



A dokumentum formája lehet rögzített, és lehet dinamikus. Dinamikus esetben a válasz alapján módosul a dokumentum szerkezete, sablonja. Szabad dokumentumok sorozata keletkezik.

Részdokumentumok alapidokumentummá (ground document) válnak, (minden változójuk értéket kapott)és átemelhetők az adatbázis címtérbe.

A kérdések és válaszok minden esetben egy adatbirtokosra tartoznak, ami több is lehet egy dokumentum életében. A bevitt új információ is adatbirtokosokhoz tartozó sémákat érint, (sémák használatára ki lehet majd térni) , mellékhatásai az adatbirtokos illetékességi körbe tartoznak. Ebből triggerek és workflow keletkezhet.

Az egész dokumentumrendszer, a lehetséges szabad dokumentumformákkal egy meta-adatbázist jelent. Ebben nem csak szerkezetek, hanem aktív komponensek is vannak, és ezt kell web-programzással létrehozni. Az aktív komponensek az interakciós lehetséges lépések programkódjai. Ebben vannak az adatbázis-kezelési kódok is. A 89-es modell elemei ide épülnek.



Az alapmodell érvényességéről:

Az információs rendszer automatizált része az alapmodell szerint jellemezhető: lehetséges adatállapotok, lehetséges adat-hozzávételek, lehetséges adatkinyerések. Mindez az algoritmos információelmélet hatókörébe tartozik.

A relációs technológia uralkodóvá válása után és a PC-k, munkaállomások megjelenése előtt a közvetlen adatbázis-kezelő felületén működött az információs rendszer.

A logikai modell kiinduló építkezése a lehetséges állapotokból indul ki. Szemantikai háttere lehet egy adatmodell, és annak bináris reprezentálása kell az algoritmuselméleti keret felépítéséhez. Az algoritmuselméleti tárgyalásra a szemantika már nem tartozik. **Ezen a szinten csak műveleti szemantika van az algoritmusok formális megadásánál.**

A logikai modell első lépésben a lehetséges módosításokkal egészül ki, a módosítások miatti zártság bővíti a logikai modellt, a lehetséges állapotok halmazát. A modell módosíthatósága, a kód optimális mérete, a homogén halmazok keresése, ezek lehetnek elméleti kérdések.

A harmadik formális összetevő az adatkinyerési kérdések és válasz-algoritmusok.

A fizikai megvalósításban mindez csak akkor működik, ha a felhasználói új adatbevitel kivételével minden tároltan van valamilyen fizikai címtartományban.



Mivel egészítsük ki a modellt, ami a mai technológiák új vonását tükrözi?

Megjelent egy közbeiktatott dokumentumforma. Mit jelent ez információelméleti szempontból? A dokumentumok képernyőbe (vagy hasonló interakciós felületre) szervezett kérdezz-felel-adatbevitel sorozatokat (időbeliség!) foglalnak algoritmusok keretébe a gépek felé, és ugyanakkor a prezentációs forma szemantikus keretet nyújt a felhasználó felé.

A dokumentumot illesztő felületként is nézhetjük: egyik arca a felhasználó felé emberi szemantika szerint érthető információ ki- és beviteli forma, másik arca a számítógépek felé adat be- és kinyeréshez adatok és eljárások azonosító adata.

A következő feladat: a dokumentum modell megadása és illesztése a két felülethez.

Néhány részlet:

A számítógép felé a dokumentum több része fizikai címekre hivatkozás. (URL például). A felhasználó felé ezek egy része a „hol és ki lépett be” keretéhez kapcsolódik, ahhoz relatív címként áll elő.

A dokumentumban szerepelnek az adatmodell elnevezéseire és a tevékenység elnevezéseire utaló részek. Ezek nagy része előre kitöltött, vagy elérési útvonalakon navigálva, keresgélve érhető el. Ezek is lekérdezések! Ezeket más adat-modellezés írja le, nem a klasszikus adatbázisok modellezése! Jönnek az ERP és hasonló építkezések.



A rendszert használó személy adott funkciójában egy szabad dokumentumot szerkeszt. Fogjuk fel először úgy, hogy a teljes vállalat egyetlen nagy dokumentumában egy részt szerkeszt. XML fogalmakkal egy adott jelölőhöz egy elemet szerkeszt. A dokumentum általa látható hierarchiáját kell illeszteni tevékenysége szervezeti hierarchiájához. Milyen adatként tárolt dokumentumokat (elemeket) milyen módon használhat? kell- lehet- engedélyezett, stb.

Milyen tevékenység (szervezeti egység) által birtokolt dokumentumokat hogyan használhat? A használt dokumentum egy alap-dokumentum, vagy származtatott dokumentum? Itt lehet tény-dokumentumok (extenzionális dokumentumok) és származtatott (intenzionális) dokumentumokkal egyaránt építeni a modellt.

A kitöltést menü-rendszer, a dokumentumok áramlását work-flow, az adatbázis, intenzionális dokumentum előállítás web-szerviz, vagy SOA rendszer támogathatná.

A cél az lenne, hogy a dokumentum egyben dokumentálja saját szerepét minden részletében. Ami keletkezik és véglegesedik, az alapidokumentummá válik. Ezt már a mai erőforrások és kezelő szoftverek tudnák kezelni.

Fontos elveket kellene keresni, például szervezeti hierarchiában lehet-e lezárás és átlépés nélkül szerkeszteni dokumentumot?

Van-e kivétel, bármi okból kívül szerkesztett dokumentum utólagos bevitele?

Hogyan épül erre a dokumentumra üzleti intelligencia?



A kiszámítás-elméleti modell továbbépítése:

A fogalmi bináris modell egy adatmodell előfordulásainak kódolása – az egyszerűség kedvéért bináris kóddá. Ebben az elhatárolók, önkorlátozó kódok, stb. használatával felépíthetők az ekvivalensnek tekinthető kódok, és egy felsorolható lehetséges állapotalmaz. Ehhez a lehetséges sémák felsorolhatósága, azon belül a véges előfordulások felsorolhatósága feltevésre van csupán szükség.

A módosító és lekérdező”metódusokat” paramétereikkel változatlanul ugyanúgy le lehetne írni. A konkrétabbá tételhez a **szabad dokumentumok** kezelésére alkalmas példákat kell megadni. A szabad dokumentumot is ebben a formában kell kezelni. A változók helye egy megengedett típusú prefix kód beillesztése. Ez XML szerkezetekre megtehető.

Két irányba kell majd tovább építeni: befele a fizikai modell felé, ez nagyjából ugyanúgy mehet a címtartományba darabolással.

Szükséges a modell kifelé építése:a szerkesztés alatt álló dokumentum felhasználói modell és a tárolt modell összekötésével. Iktatás és hasonlókérdésekre is legyen kifejezési lehetőség.

A három adat



1. Az adatmodell és előfordulása: a valós „tárgyak” gyűjteményei, érthető halmazok és tulajdonságok, viszonyok megadásával.
2. A dokumentumok gyűjteménye: ami tevékenységek (emberi és instrumentumok) működése során az adatmodell tárgyaira is ható változás, és dokumentálódik.
3. A „képi” észlelések adatai, amire szemantikai feldolgozás épül, és új felvétel illeszthető korábbiakkal, kijelöléssel kérdéssé, dokumentálássá alakítható.

Adatmodell és Dokumentummodell



Az adatmodell a szervezet (tárgyrendszer) számára a létezőnek tekintett valós világra vonatkozó tények, összefüggések rögzítése – időfüggéssel. A modell (gyűjtemények felépítése) egyben egy szemantikai referencia is! Az adatmodell adatai speciális, kitüntetett adatok. Fogalmi adatmodellként kell először tekinteni! **Az előző ábra 1. és 3. típusú adatainak adatbázisa.**

A dokumentummodell a szervezetben végbemenő változások dokumentálása.

A változásokban részt vesznek az adatmodellben szereplő valós-világ elemek (entitások), ezek adják a dokumentum szemantikai környezetét. A dokumentumban egyértelmű hivatkozás, azonosítás kell ezekre.

A dokumentummodell a szervezet tevékenységét szabályozott formában követi. A dokumentum áramlik (pl. adatáram-diagramok), ezért át-, ki- és belép szervezeti határokon, információként követi, mi történik. Közbeiktatható két tevékenység közé, mint adat tud függetleníteni (izolálni) tevékenységeket.

Adatmodell és Dokumentummodell



Az adatmodell kezelése az adatbázis-kezelő rendszerre tartozik.

A dokumentummodell mai szabvány leírása lehet XML. Amit névtérrel lehet egybekapcsolni az adatmodellel.

A névtérben megnevezhetők a szervezeti egységek, berendezések, stb, ami a változásban szerepet kap. Ezek egy része a dokumentum mindenkori kitöltési helyéhez kötve automatikussá tehetők. Így a dokumentum szerkezete követheti a szervezeti tagolódást, a változásban érintett entitások tulajdonosi és egyéb viszonyai is megjeleníthetők.

A dokumentum megjelenési formája: a személy funkciójától is függő lehet, Lehet aktív, passzív és véglegesített, részdokumentum, instrumentum dokumentuma, stb.

A személyi felületek lehetnek web-típusúak, browser-en keresztül, web-szolgáltatáson, SOA-típusúak, vagy SAS típusúak. Ezek a felületek kötik össze a dokumentum informatikai feldolgozó algoritmusai az adatbázisban tárolt tény-adathalmazzal. Tipikusan új dokumentum kitöltését kezdeményezhetik, ami egy kijelölt változásról készül majd.

ABKR



Megszorítások:

- Mi lehet az adatbázis tartalma?
- (valamilyen formalizálni kell)
- Milyen új üzenetek lehetségesek;
- Hogyan épülnek be a régi ismeretekbe?
- Milyen kérdések fogalmazhatók meg?
- Mik a válaszfüggvények?
- (tömöríthetőség, hatékony lekérdezhetőség –
ellentmond a módosíthatóságnak)

a) Mi lehet az adatbázis lehetséges állapota



$\Sigma \subseteq \underbrace{\Omega}_{\text{véges bináris szavak}}$ Σ : rekurzív v. rekurzívan felsorolható

Sémák használata : sémaleírás = adat, tehát :

$S \subseteq \Omega, (\omega \cup \Omega \text{ tetszőleges bináris string})$

$s \in S, \Sigma_S$: séma szerinti lehetséges adatok (bináris kódok)

$\langle s, y \rangle = x = s'y$ (S séma beli y előfordulás)

s_1, \dots, s_n sé mák

y_1, \dots, y_n állapotok

$\langle s_i, y_i \rangle$ felsoroló kód



b) Hogyan lehet módosítani?

Y : módosító adatok halmaza

$M : Y \times \Sigma \rightarrow \Sigma$ módosító függvények halmaza (kiszámítható függvény)

$m \in M, y \in Y : m(y, x) = x'$, y lehetséges módosító adat

zártsági elvárás :

$$M(Y \times \Sigma) \subseteq \Sigma$$

szemantika az adatmodellben dől el

\Rightarrow generikus ekvivalencia - osztályok Σ - án

olyan legyen a módosítás, hogy az eredmény értelmezhető legyen

pl. ekvivalencia - osztályok bevezetése Σ - án

üres módosítás : $\Lambda, x = x', ha$

$$M(\Lambda, x) = M(\Lambda, x')$$

sémánkenti módosítások



b) Hogyan lehet módosítani?

sémánkénti módosítások :

$s \in S, \sum_s, M_s \subseteq M$, módosító függvények halmaza

Adat mennyiségek :

-hosszal mérhetjük - redundancia

- $K(x), C(x)$

Ezek alapján elemezhető a módosítások
komplexitása

Séma megválasztása mikor jó?

(Kolmogorov bonyolultság alapján)

$x \in \underbrace{\sum_s}_{\text{véges}}, K(x|S) \sim \log_2 N, |\sum_s| = N$

\sum_s – en egyenletes kódot használhatunk

\sum_s – en homogén feladat lesz

b) Hogyan lehet módosítani?

Reláció séma



$$S = R(A_1, \dots, A_n),$$

$$A_i \rightarrow |D_i| = m_i \text{ (domain, értéktartomány)}$$

l - az S előfordulása

$$\text{Lehetséges előfordulások száma: } 2^{\prod_{i=1}^n m_i} = |\Sigma_S| = N$$

$$\text{sorok száma: } \prod_{i=1}^n m_i = m$$

Azonban a tipikusfeladat

$$\text{legfeljebb } k \text{ sor van benne: } N - \binom{m}{k} \approx m^k$$

$$\text{egyenletes kód: } k \cdot \underbrace{\log_2 m}_{\text{sorokénti kód}}$$

minden lehetséges függőség korlátozza a sorok számát \Rightarrow

tömörebb ábrázolási lehetőség

(Soroként kódolják a táblákat a rel. adatbázisok)



c) Lekérdezések megfogalmazása

Q : a kérdések halmaza

$q \in Q : \Sigma_q$: a válaszok halmaza

típusos v. jól tipizált kérdésekkel foglalkozunk

\exists válasz fv. q -hoz

$A: \underbrace{Q}_{\text{kérdés}} \times \underbrace{\Sigma}_{\text{állapot az adatbázisból}} \rightarrow \bigcup_{q \in Q} \Sigma_q$: válasz függvény

$q \in Q; a \in A: a(q, x) = v \in \Sigma_q$ (értelmes)

Lehet, hogy a kérdésnek nincs értelme, de azt tudom detektálni.

Lehet, hogy (a válasznak?) nincs értelme, de ezt nem tudom megmondani

kérdések bonyolultsága \Rightarrow generikus

generikusság: ekvivalens állapotokból, ekvivalens válaszokat kell kapni.



c) Fizikai modell

Fizikai modell :

-bedaraboló dik egy bináris kód

címezhető véges kapacitású rekeszekbe

Ha van egy fizikai kód :

$$\left(\underbrace{c_1}_{\text{cím}}, \underbrace{l_1}_{\text{hossz}}, \underbrace{y_1}_{\text{adat}} \right), \dots, (c_n, l_n, y_n)$$

φ - fizikai kód, $z \in \Phi$

$$f : \Sigma \rightarrow \Phi, \quad x \in \Sigma, \quad \varphi_x \subseteq \Phi$$

Módosító k rendszere : $y \in Y, \quad \varphi_y; \omega \in \varphi_y, z \in \varphi_x$

a sz.gép megcsinálja a,
implementálja a változást

$$\underbrace{f_m}_{\rightarrow}$$

$$\varphi_{x,y} = \varphi_y \circ \varphi_x =$$

$$\underbrace{\omega z}$$

$$v \in \varphi_{m(x,y)}$$

diszjunkt tartományb an legyenek

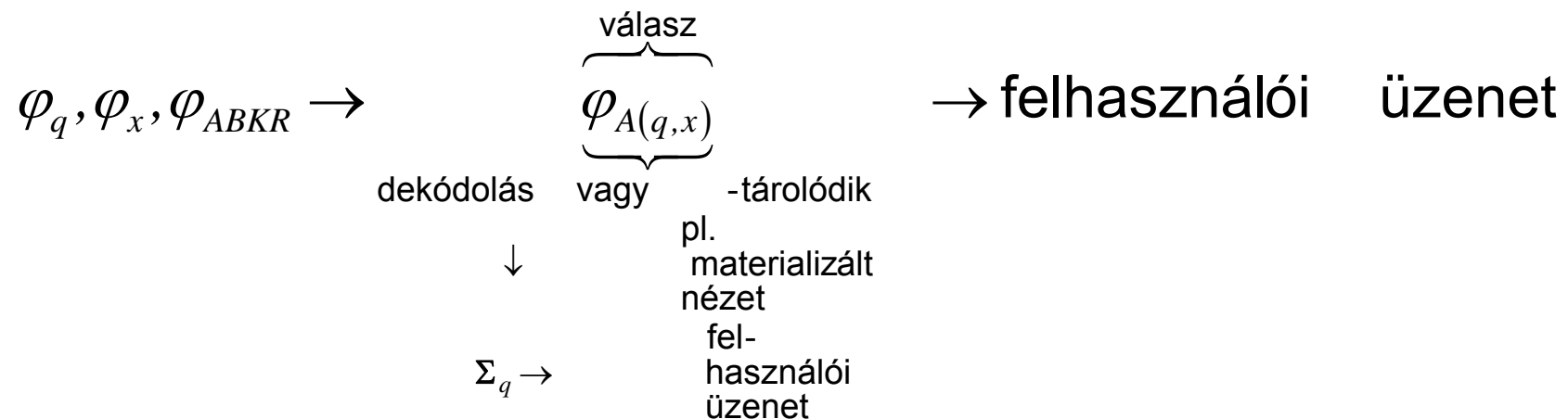


c) Fizikai modell

Eljárásainknak is lesz egy kódolt formája
ugyanakkor a címtartományban mozog.

Lekérdezés:

$$-q - k \in \varphi_q,$$



Matematikai kitérő –

