



Információs rendszerek elméleti alapjai

Információelmélet

Planning and Design of Information Systems



André Blokdijk, Paul Blokdijk

ACADEMIC PRESS, 1987.



Bevezetés

Információs rendszer minősége - tervezés hatékonysága

Módszerek keletkezése - 70-es években, tapasztalat és elmélet kölcsönhatása

kell elmélet sikeres gyakorlati alkalmazáshoz

kell elméleti teszt, mi ködik-e majd a technika, módszer a gyakorlati alkalmazásban - elkerülni a megjósolható hibákat

sikertelen technika, módszer esetén új elméleti háttérre lesz szükség

elmélet: megelőzni a problémákat, mielőtt el fordulnak.

Információs rendszerek igen sokféle környezetben működnek, sikeres tervezésükhöz kell elméleti tudás és megalapozás.



Terminológia és osztályozás (taxonómia)

Fogalmak tisztázása:

Módszertan - módszerek elméleti vizsgálata

Módszer - szisztematikus eljárás, technika, kérdezési -
interjú mód egy elvi szemléletmódhoz tartozóan,
fogások; technikák rendszere

Technika - eljárása, törzse egy technikai lépésnek

Leírási technika - jelölésrendszer

Osztályozás:

Tudományos paradigma: adatanalízis + strukturált módszer

Rendszer paradigma: általános rendszerelmélet + emberi
aktivitás + részvételi megközelítés



Filozófiák:

Projektvezetés, mint kiindulás: kézikönyvek, el írt lépések,..
Rugalmatlan, jó a vezetésnek

Adat, mint kiindulás: adatmodell, normalizálás, I-O, lekérdezés, de
nincs szervezet a háttérben

Funkcionális elv kiindulás: funkciók hierarchiája, I/O viszonyok ,
de nem ad rendszer-részrendszer felbontást

Rendszer elv kiindulás:

Rendszer= kapcsolatban lévő entitások halmaza, nincs
részhalmaza, amely ne kapcsolódna más részhalmazához,
kapcsolata van környezetével.

Entitás: elkülönül , szeparált létezik a tárgyi-eszmei világból

Fekete doboz elv



6 Planning and Design of Information Systems

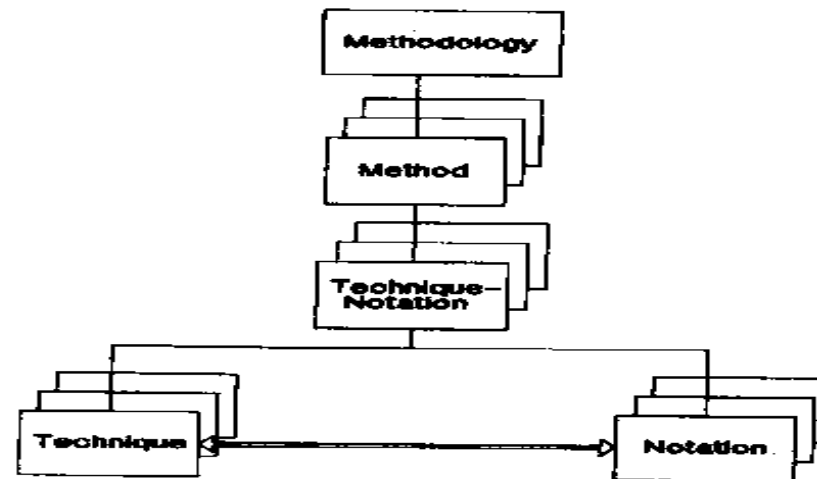


Fig. 1. Methodology — Method — Technique — Notation

For a long time we have used the classification in Figure 2 in studying other methods.

If we compare these taxonomies, we find some similarities and overlap. General systems theory and systems thinking can be used in the human

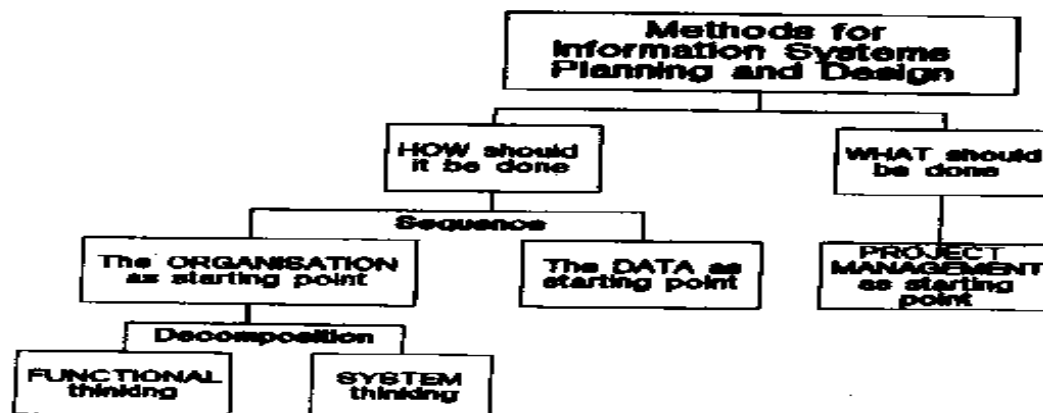


Fig. 2. Taxonomy of Methods



Módszertan alapkérdései

Megadni, mit kell tervezni és implementálni egy módszer használatával?

1. Axióma: mindazt, ami megváltozik a valós világban, a szervezeti tevékenységek automatizálása miatt.

Tapasztalati evidencia, mi változik.

I. - A munkavégzés módja (szervezetben és általa)
- A munka felosztása
- Az együttm ködés szerkezete

2. Axióma: A szervezeti tevékenységek automatizálása egy átszervezési aktivitás. Az automatizálásnak mindig van átszervezési vonzata.

A szervezeti tevékenységek tervezése nem hagyható ki!



II. A szükséges információ is változhat.

Információ - ami egy feladat végrehajtásához, irányításához/ellenőrzéséhez, benne/felette való döntéshez szükséges.

Információ: ami eléri az ember tudatát és hozzájárul tudásához.

Káosz - maximális entrópiára törekvés, az információ ezt küzdi le.

Tudni kell: miért, mit, mikor, hogyan

Tudás: a megfelelő tevékenység megfelelő időben való végrehajtásához.

III. A szükséges adatok változhatnak

A tárolt, szolgáltatott, feldolgozott adatok szinte mindig változnak.

Adat - a ma tudása a jövő számára.

Információ-hozzáadás: mindenre, ami előző információ volt, szükség van.



Számítógépes adatok - tárolás és feldolgozás emberi korlátjait oldja fel.

Információ - adat - információ körforgás

Munka integrálás \Rightarrow adatintegrálás, fordított irány nem egészséges!

IV. A szervezeti tevékenységekben használt eljárások változhatnak.

- a tevékenység, eljárás tartalma
 - feltételei, körülményei, ahogy végrehajtnak
 - bels feltételei, döntései
 - folyamatok szerkezetei, egymásutániség, hívás, iteráció, stb.
 - a végrehajtó személye

3.0 Methodology

As a methodological issue in the context of this book, we should define *what must be designed and implemented using a method*. Starting from the *axiom that all that changes in the real world through the automation of organisational activities should be designed and implemented*, empirical evidence shows us what is changed.

Executing and observing many planning and design projects, we found that one or several of the following aspects can be changed through automation.

1. The method of working in or by the organisation is changed

- from non-automated to automated work, the computer instead of the human being, and/or
- from batch, postponed processing, to on-line real time processing of the work, and/or
- from a non-integrated, sub-optimised to an integrated optimised way of working in the organisation.

The timing and the quantities of the work must or can be changed.

The division of working

- from one task to another new designed task,
- from one functionary to another functionary,
- from one department to another department.

The structure of the cooperation

- functions are moved from one department to another,
- reporting lines are changed, or new ones are established,
- management styles change: control, decision making, people management.

This leads to a second *axiom: the automation of organisational activities is a (re)organisation activity in itself, or: automation always has reorganisation implications*. This means that the organisational activities design activity may not be neglected in any information systems design and development method.



Modellek

Az automatizálás keretében fellépő szituációk elemzése, tervezése modellt igényel. A jelen/jövő szervezeti tevékenységeinek szituációihoz.

Miért kell? Munkaerő képesség, megvalósíthatóság ellenőrzéséhez, bonyolultság kezeléséhez. Nem lehet mindent előre teljes implementálási részletességgel tudni. A modell: közbülső lépés.

Mire? **A 4 fő változásra.** 4 fő aspektushoz kell modell:

szervezeti modell, ami a valós tevékenységek (munkák) felosztását, módját mutatja

információ modell, az információ tartalmát, és származtatását mutatja (a valós világ viszonyai szerint, ami a tevékenységekhez kell)

adatmodell; mutatja a valós objektumokat, amelyekről információkat tárolunk, az objektumok viszonyait

eljárás modell, a valós tevékenységek struktúrája, irányítása fejeződik ki benne.



Ezek a modellek összekapcsolódnak, integrálódnak explicit/implicit módon, a lehető legkevesebb átlapolással, redundanciával.

Az aspektusmodelleknek 3 absztrakciós szintje van:

1. A fogalmi (konceptuális, logikai) szint
2. Technikai-fizikai szint (ahogy a használatban megjelenik.)
3. Implementációs szint

A 3-szor 4 modell vázlatosan:

1. Fogalmi szint: absztrakt, de a használónak érthető, nem technikai szint. Milyen lesz a valós világ a jövőben?

A) szervezeti modell: milyen szervezeti tevékenységek, mi köti össze őket (anyagi-információ értelemben)

B) információ modell: milyen információ kell a tevékenységekhez, mi a származtatásuk, forrásuk



- C) adatmodell: milyen adatstruktúrát kell tárolni, szétosztani, melyek az összefüggések az adatok között
 - D) eljárás modell: az egyes tevékenységek által végrehajtott eljárások tartalma, struktúrája; hogyan kell a tevékenységeket megkövetelni, ellenőrizni (üzleti és jogi szabályok)
2. **Technikai szint (fizikai)** ami számítógépesítésre kerül, azt technikai, fizikai fogalmakkal kell leírni
- A) szervezeti modell: gépi tranzakciók, gépesített tevékenységek programja, összekötési hálózat a szervezeti egységek között, a nem gépes tevékenységek algoritmusai
 - B) információmodell: az információ megjelenési formáinak tervei (kibocsátott és kapott információra egyaránt)
 - C) adatmodell: ABKR terminológiájú adatmodell leírás
 - D) eljárásmodell: programtervezés szint (input-output), + kommunikációs eljárások



3. Implementálási szint: ahogy megvalósul: számítógépek, terminálok, hálózat, fájlok, adatbázis, stb.

A) Szervezeti modell:

tevékenységek: -feladatok a személyek funkcióihoz

-funkciók szervezeti egységekhez

- egységek a jelentési rendszerben

- felhasználói eljárások feladatonként

- kommunikációs hálózat

B) Információmodel: deklarációs forma jelleg specifikációk, menük, stb.

C) Adatmodell: a fizikai adattárolás reprezentációs szintjén

D) Eljárásmodell: programozás-nyelvi fogalmakkal

Systeological - Infological - Datalogical - Technological
megközelítések

14 Planning and Design of Information Systems

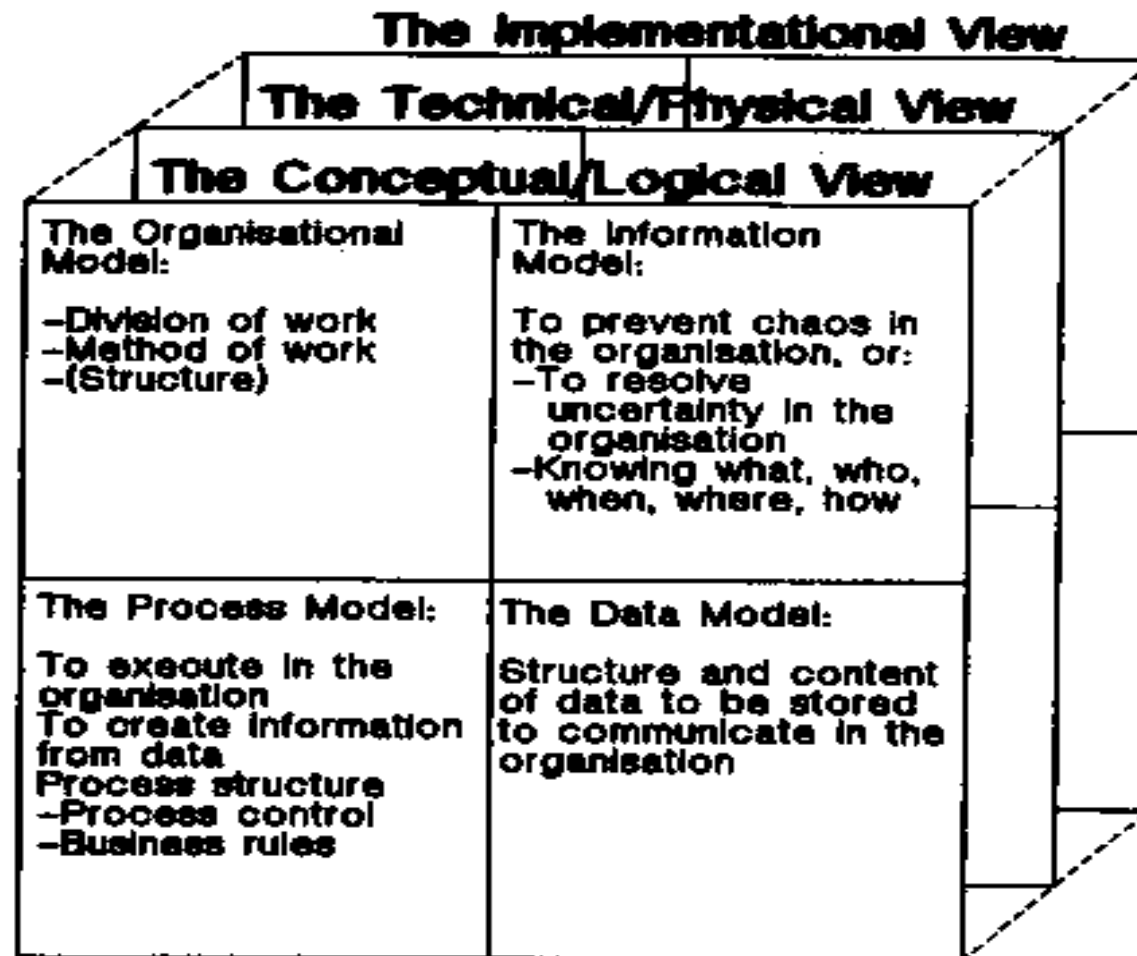


Fig. 4. Four Models in Three Views

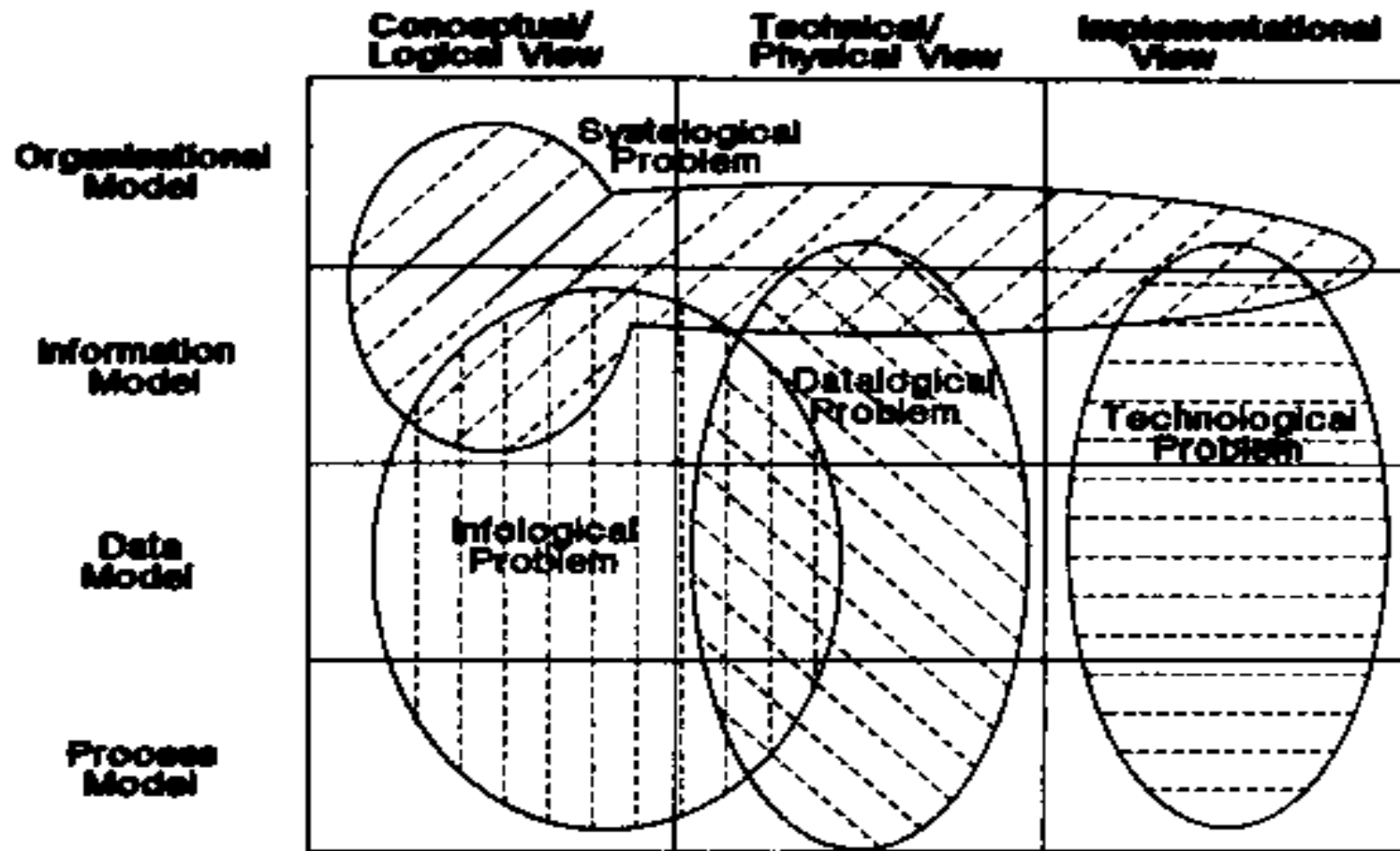
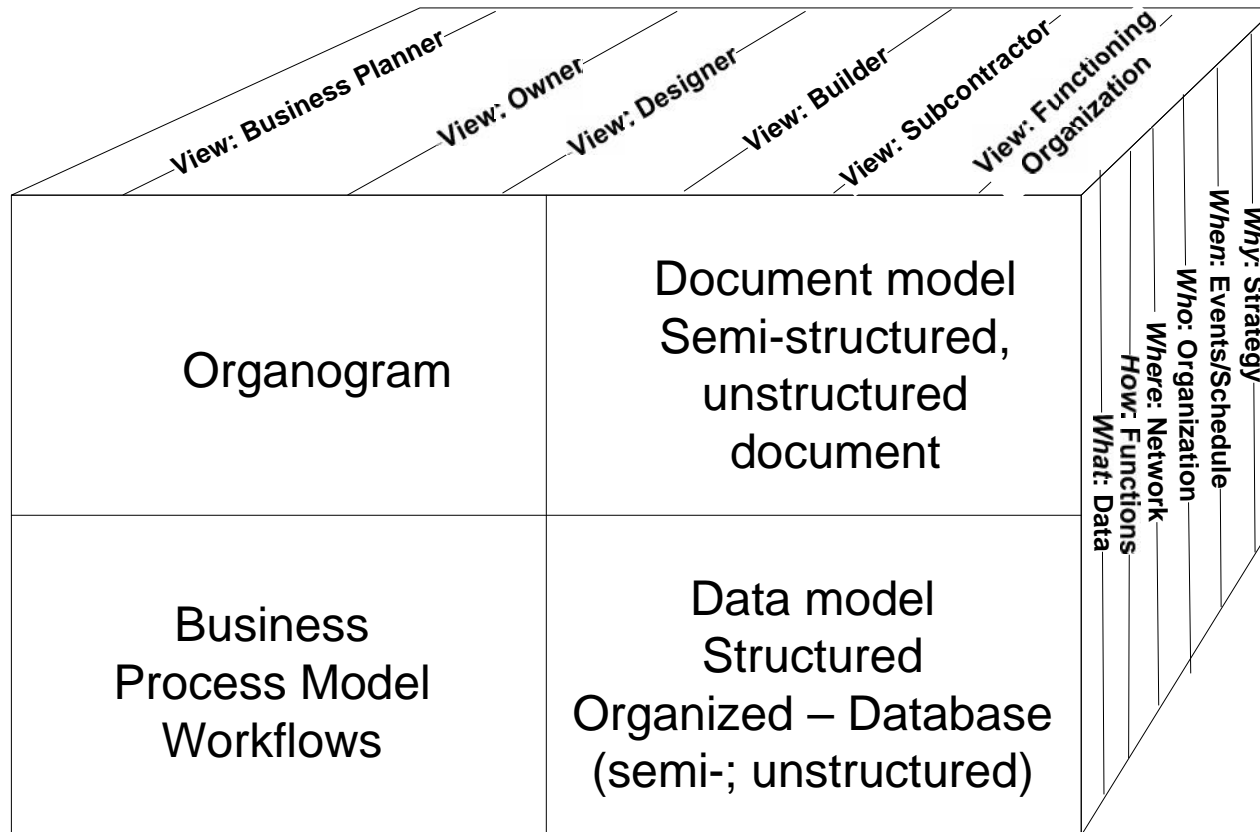


Fig. 5. Models, Views and the Problems





3.3 A tárgyrendszer és részrendszerei

IR önmagában nincs. Ált. szokás: azt, amit az IR kiszolgál, tárgyrendszernek nevezzük.

(Langefors: az irányított, vezetett rendszer = tárgyrendszer.

Nissen: az IR célja: egy nagyobb, egész rendszer kiszolgálása - ennek részrendszere - a kiszolgált rendszer a tárgyrendszer, TR.)

Mi a részrendszer?

Def: felismerhet része egy rendszernek, elkülönül más részeket l, és mind a részen belül, mind a többi résszel való együttm ködésben minden kapcsolata megvalósul.

Részrendszer integritása - minden kapcsolata m ködik; ezt a technikák, jelölések állandóan ellen rzik!

Tárgyrendszer tartalmazza az IR-t. Van egy formális, és egy informális IR. Az utóbbi nem automatizálható, informális szervezet.



Számunkra érdekes: a formális információs rendszer. De új IR tervezésénél az informálisból is formalizálódik valami!

Def: IR (Langefors) : (1) információ készleteknek olyan rendszere, amelyek egy nagyobb rendszerben a döntésekhez és jelzésekhez szükségesek, (2) tartalmaz alrendszereket az információ készletek gyűjtésére, tárolására, feldolgozására és szétosztására. (A nagyobb a TR)

További részrendszerek :

formális IR: - automatizált és nem automatizált adatfeldolgozó rendszer

automatizált adatfeldolgozó rendszer:

- adatrendszer

- operációs rendszer

- hardver rendszer

Rendszer(elv) gondolkodás: gyűjtés kintre | befelé.

Ábrák: rendszer ábrázolás, gyűjtés k.



Győr elv és aspektus rendszerek

Egy aspektus-rendszer cseréje - rendszerbe illesztést igényel mert nem változatlan minden kapcsolata. (Szétesés adalékokra - aggregátum)

Az IR nem tekinthet önmagában lévő valós rendszernek.

Aspektus változása: IR változása a teljes TR rendszer nélkül nem lehet! IR tervezése alárendelődik TR tervezésének!

Győr elv és a 12 modell viszonya:

TR: a 4 fogalmi szint

formális IR: a 4 technikai modell

automatizált adatfeldolgozó rendszer: a 4 implementációs modell

3.4 Modelljeink a jövő modelljei. **Jelen és jövő tervezési viszonya:**

Rossz: jelen túl részletes felmérése. Időigényes, nem ebből kell a jövőt levezetni!

1. Mi kell a jelenből: megérteni az egészet és nem a részleteit; azok ugyanis változnak.



2. Az automatizálás célja: megújítani a szervezeti tevékenységet a számítógépesítéssel. Jelen részletezéséből csak javítás lesz.
 3. A csak javítás, kis lépés - nem lesz arányos a haszon a befektetéssel
 4. Hamar szükségessé válhat egy újabb újratervezése az IR-nek, minden változás többletteleher a felhasználói területen.
 5. Ha mégis kell az újhoz a jelen, elég lesz akkor megvizsgálni.
 6. A jelen elemzése a tervezés szempontjából lehet fontos, hogy „jól megértse a jelent” az új tervezéséhez. Ismerni kell a háttérrel, de nem jó alap a régiekkel való kiindulás.
- Vigyázat! Túl nagy lépés előre nem jó a felhasználónak! Nem látja a lehetőségeit az automatizálásnak, nincs innovatív terv-cél, amiért harcolni kell.
- Szokásos innovációs stratégia: a felhasználók bevonása, tanítása a tervezés folyamatában. (Külön fejezet a könyvben.)

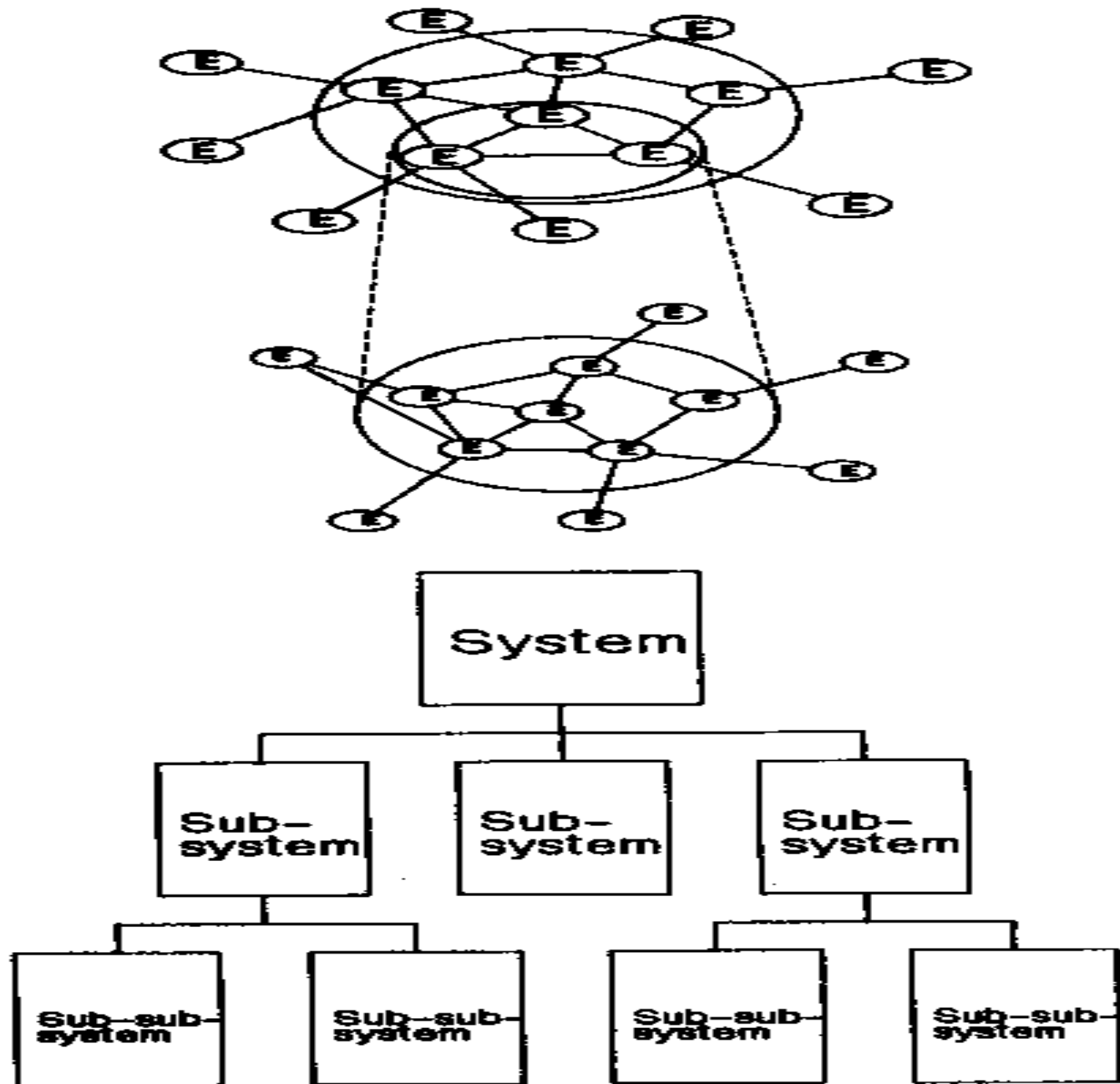


Fig. 6. Two Notations to Represent Subsystems

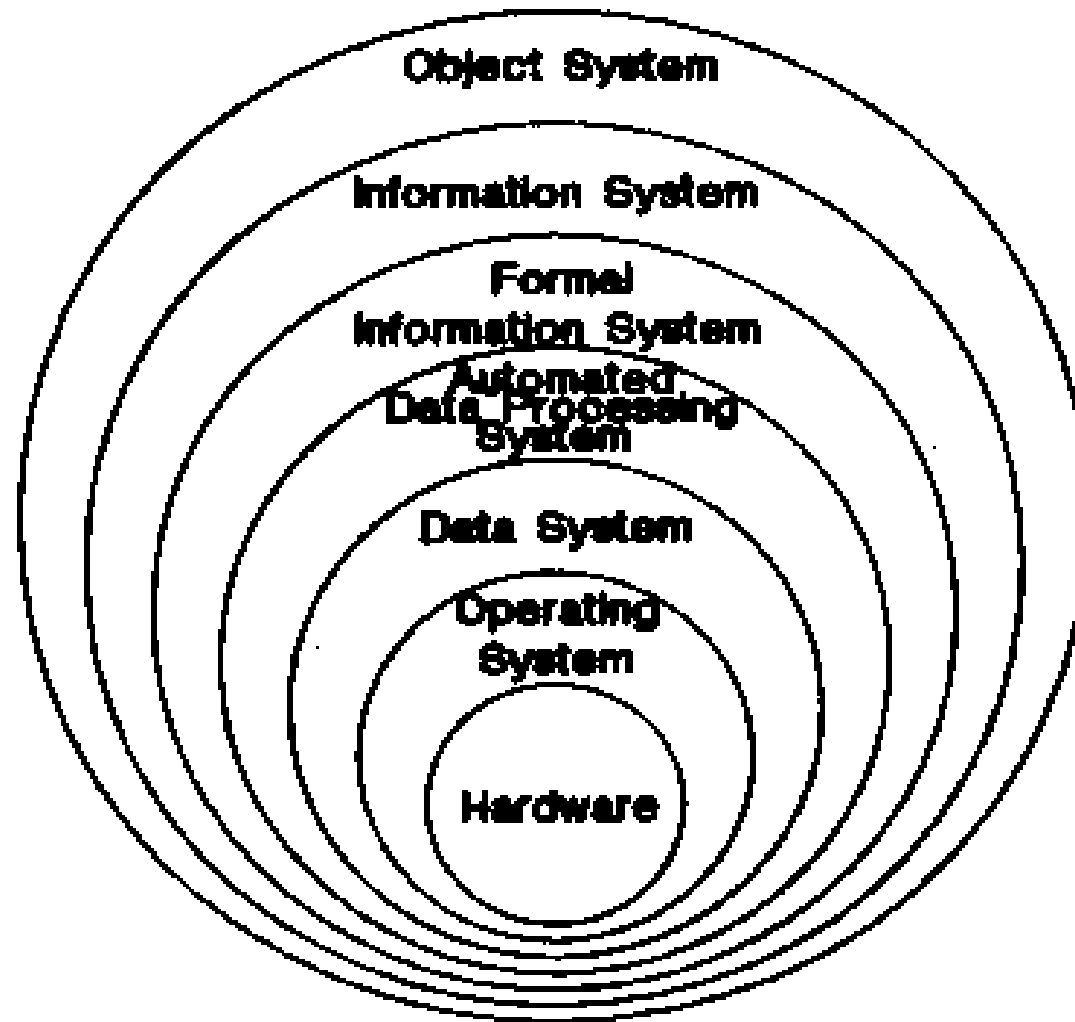


Fig. 7. Subsystems of the Object System: 'Rings Concept'



3.5 Aspektusok sorrendje

Mivel kezdjük? A 4 modellre több érvelés született:

- Az adat a legstabilabb. Kezdjük ezzel.

Viszont a szervezeti tevékenység változása kihat az adatokra is!

- A szervezeti tevékenységek, munkakörök kijelölik a szükséges adatok körét. De mi van a környezeti adatokkal, az információs rendszerrel kommunikált adatokkal?

- Adatintegrálás \Rightarrow szervezeti integrálás, tehát először adatmodell. (70-es években tipikus volt). Eredmény: egy monstrum adatbázis az egésznek. Túl nagy, nehéz implementálni, nem praktikus.

- Információ az egész alapja - először információ modell. De miről, minek az információ? Az info. Célja: segítsen a szervezeti tevékenységekben/-ről a döntésekben, ellenőrzésben, végrehajtásban.

- Eljárások adják a szervezeti tevékenységek dinamikáját. Először eljárás modell. De mit dolgoznak fel? Adatot, anyagot, stb., nem kellene előbb tudni mit?



Sok eszme, érv, ellenérv. De mi van a felhasználóval?

Mi érdekli? A munka hogyan osztódik fel, hogyan kell végezni, ki irányítja, hogyan irányítja?

A min ségi szempont: a legfontosabb a szervezeti modell, hogy jól kielégítse a felhasználókat. Tehát: els a szervezeti modell!Utána a sorrend függ a rendszer típusától.

Gazdasági, vállalati rendszer: szervezeti - információ - adat - eljárás

Információszolgáltató: szervezeti -adat - információ - eljárás

Termelésirányítás, gyártásirányítás: szervezet - eljárás - inf. - adat

3.6 A komplexitás leküzdése

Gyakori hiba: a komplexitás kezelésének képtelensége.

Langefors: Felfoghatatlan rendszer: a részeinek száma és kapcsolatainak száma olyan nagy, hogy a teljes struktúrájuk megbízhatóan nem fogható fel, vagy nem figyelhet meg egyazon id ben.

3 út: Top-down, Bottom-up, Centre-out

Tudni kell, mikor melyik jó.

There are three ways for handling complexity:

- **Top-down:** starting with one model of high abstraction, e.g. on one A4 sheet of paper, and breaking it down, decomposing or refining it, until the required level of detail is reached.
- **Bottom-up:** starting from the most detailed level and synthesising those details step by step to a higher level of abstraction, until one single model on an A4 sheet is obtained.

Both work with the concepts of subsystems or subfunctions.

- **Centre-out:** this technique is used on *one* level of detail, for example the user task. Starting from one important user task, the adjacent interfacing tasks are found and defined, until the boundaries of the information system are reached (see Figure 8). Every circle is on the same level of detail. It is applied in the organisational, information and process model.

For every step in a method it should be clear which strategy is used, and why this is the best solution for this step. A mixture of these strategies will be seen over all steps in a method.

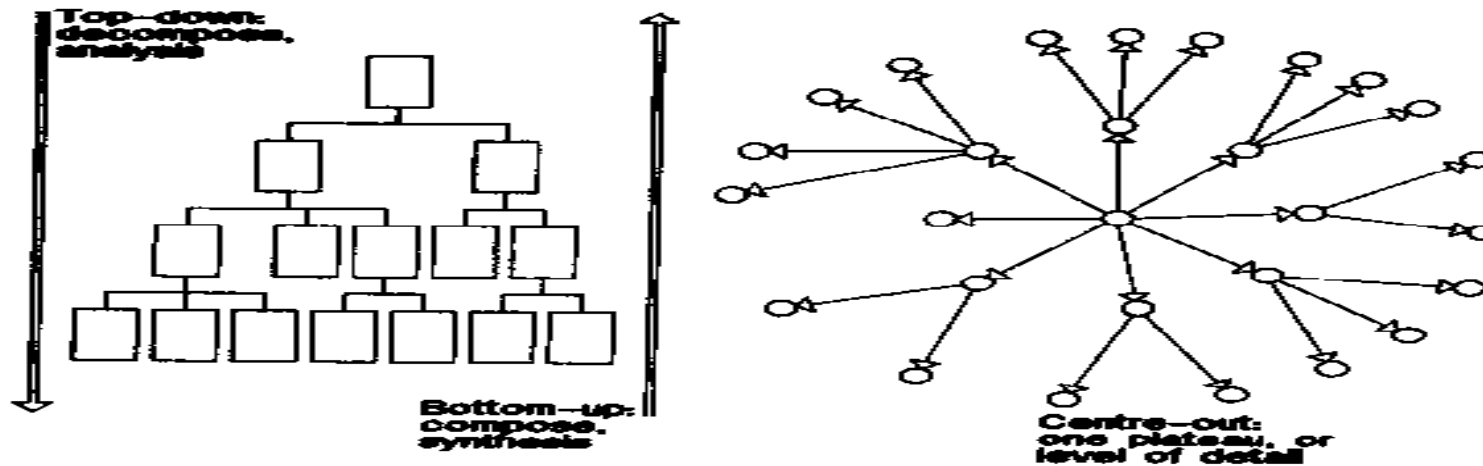


Fig. 8. Top-down, Bottom-up and Centre-out Planning and Design

3.7 OBJECT SYSTEM AND ITS SUBSYSTEMS (2)

As we have just stated, both the top-down and the bottom-up strategy also have something to do with the notion of subsystems of the object system,



3.7. A tárgyrendszer és részrendszerei

- hierarchia - foglalkozik részrendszerekkel
- aspektus rendszerek - foglalkoznak részrendszerekkel

A tárgyrendszerben vannak még tematikus részrendszerek.

9. Ábra - tortaszeletelés elve

Kapcsolat a képen a részek között nem látható, de érezhető. A háztartás (House) tovább szeletelődik.

10. Ábra: befele még egy gyűrű, az információ rendszer. Nem a tárgyak, hanem az őket érintő információ.

11. Ábra: tovább befele.

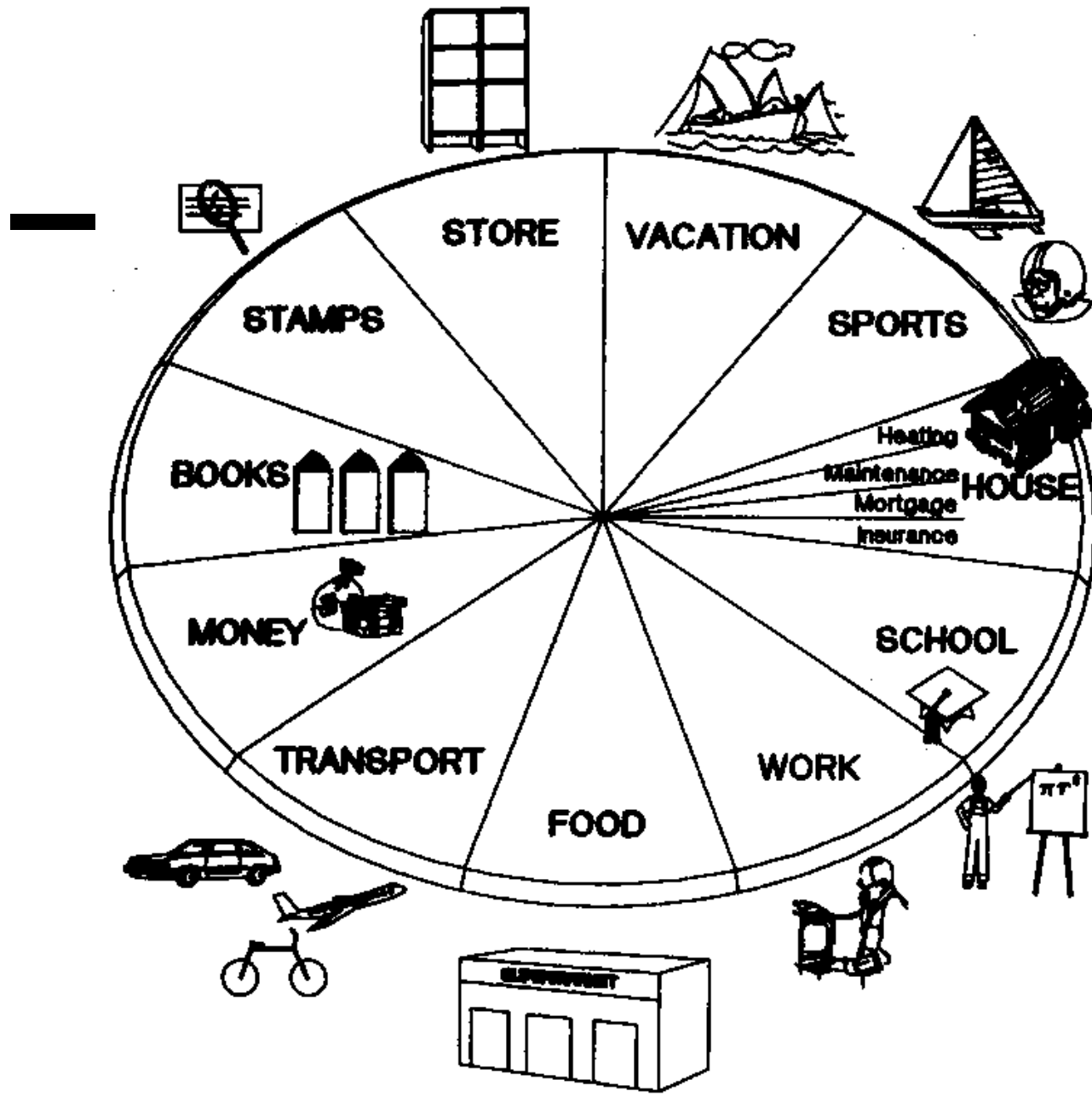


Fig. 9. Subsystems of the Object System: 'Wedge-of-Cake Concept'

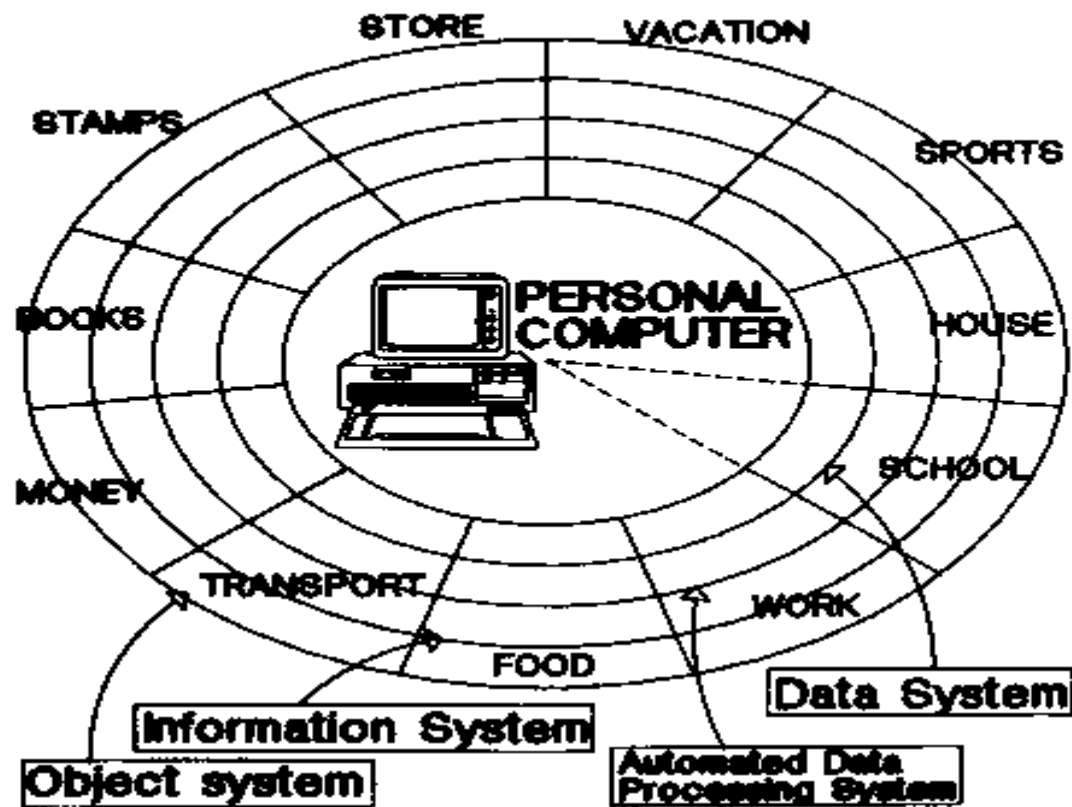


Fig. 11. 'Wedge-of-Cake — Rings Concept': Other Subsystems

Starting from the four models needed to design completely an object and an information system, we have seen so far five methodological issues:

- the three views on the four models;
- the object system and its subsystems;
- descriptions of the present and/or future situation;
- the sequence strategy of these aspects;
- complexity controlling strategies.

There are more methodological issues which influence the planning and design practice; they will be discussed hereafter.



4. Technikák, jelölések, határok (korlátok), választások

4.1. Tervezési technikák

Sok technika létezik, ismert. Minden készül modellhez kell egy megfelelő tervezési technika, amivel a modellezés elvégezhető. Egy technika: sajátos szituációhoz, sajátos modellhez, sajátos gondolkodásmódhoz illeszkedik.

- **precedencia (előzmény) analízis:** (egy absztrakciós szinten, ált. T-D része). Kiindulás: a kimenet eredmény, mi kell előállításához, mit kell tenni. Outputból a tevékenység input igénye.
- **succedence (követési) analízis:** fordított irány
- **funkcionális dekomponálás, strukturált tervezés**
dekomponálás alapja: forrás - átalakítás - kibocsátás (input - process - output), legfontosabb a funkció - részfunkció közötti áramlás. T-D módszerhez. Szervezeti és eljárásmodellhez jó.



- **Grid-chart (keresztábla)** - két aspektus kapcsolatainak mátrixa fontos ellenőrzési technika sok módszerhez. 14. Ábra 4 aspektusra.
- **Adatszintézis** - fogalmi adatmodell több felhasználói modellből, kapcsolat, származtatás-kötés, adatok között
- **Adatnormalizálás** - relációs adatmodellben, 3-5NF, BCNF
- **Strukturált eljárások** (Dijkstra) - szekvencia-elágazás-iteráció
- **komponens analízis** (Langefors): információ és adat felosztása részekre egészen az elemi üzenet, adat szintig

4.2. Tervezési jelölések

16.- 23. ábrák

- ***Precedence analysis:*** This technique, originated by Langefors (1973), works on one abstraction level and is mostly applied in the top-down strategy. Starting from the definition of the required results for the environment, an analysis is made of what should be done and what is needed to deliver that result, until the deliverables from the environment are found. In short: from the output to the input of an information system or a part of it (see Figure 12). It is a technique used in the systems thinking approach for the organisational model and the information model.
- ***Succedence analysis:*** This is the reverse of the previous technique, from

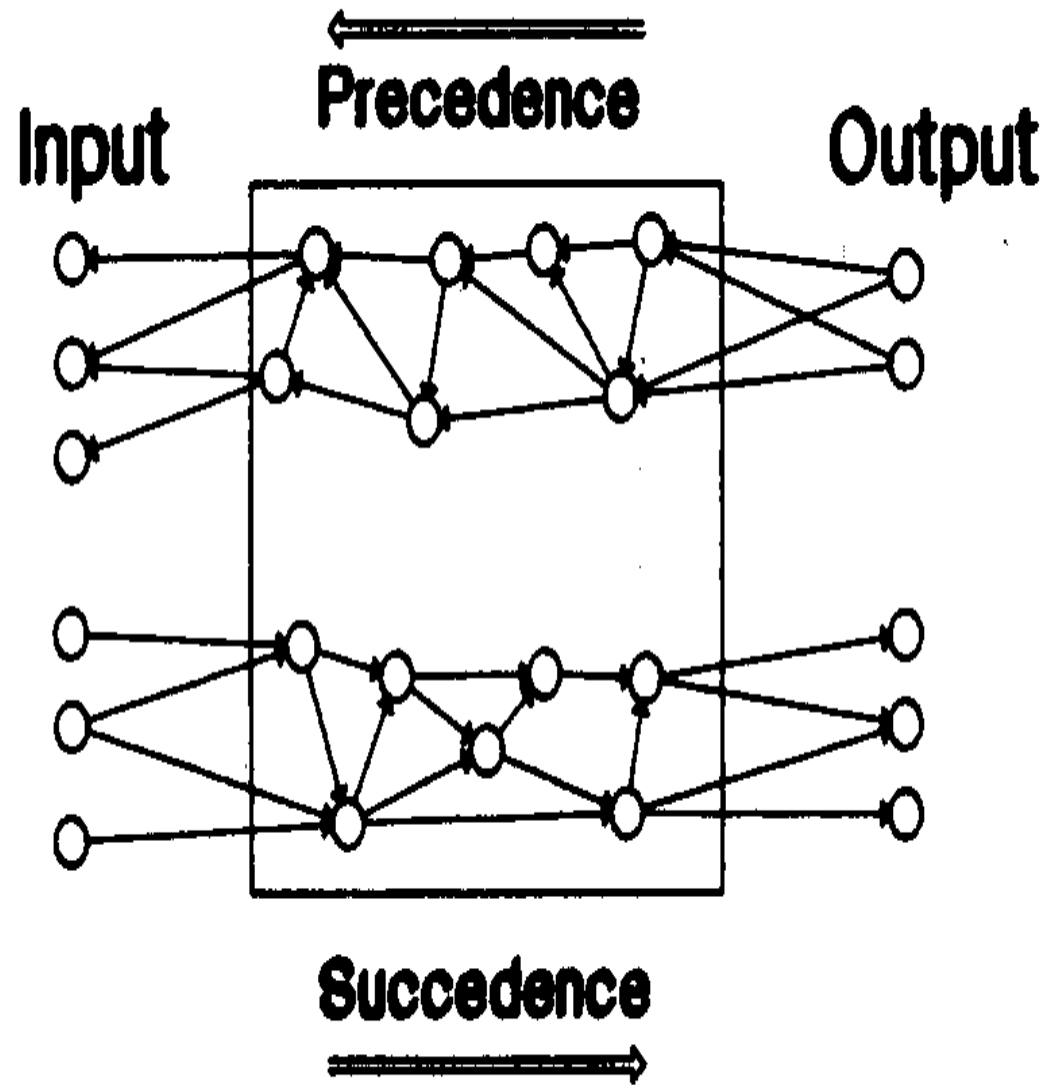


Fig. 12. Precedence and Succedence Analysis



28 Planning and Design of Information Systems

the same origin. This works from input to output: what is done with this input and what the results are.

- ***Functional decomposition and structured design:*** This technique decomposes functions according to the source, transform and sink distinction. This means that input, process and output are the guidelines for decomposition or structuring. By defining what is passed over from function to subfunction, the input and output of a (sub)function from and to a higher level function are defined. It is used in a top-down manner. See Figure 13. While in the precedence analysis technique *abstraction* levels are used, i.e. no real world processes have to be executed in the abstraction levels, here all the (sub)function levels can contain real world processes. It is used for the organisational and process models. Stevens et al. (1974) are the originators of this technique.

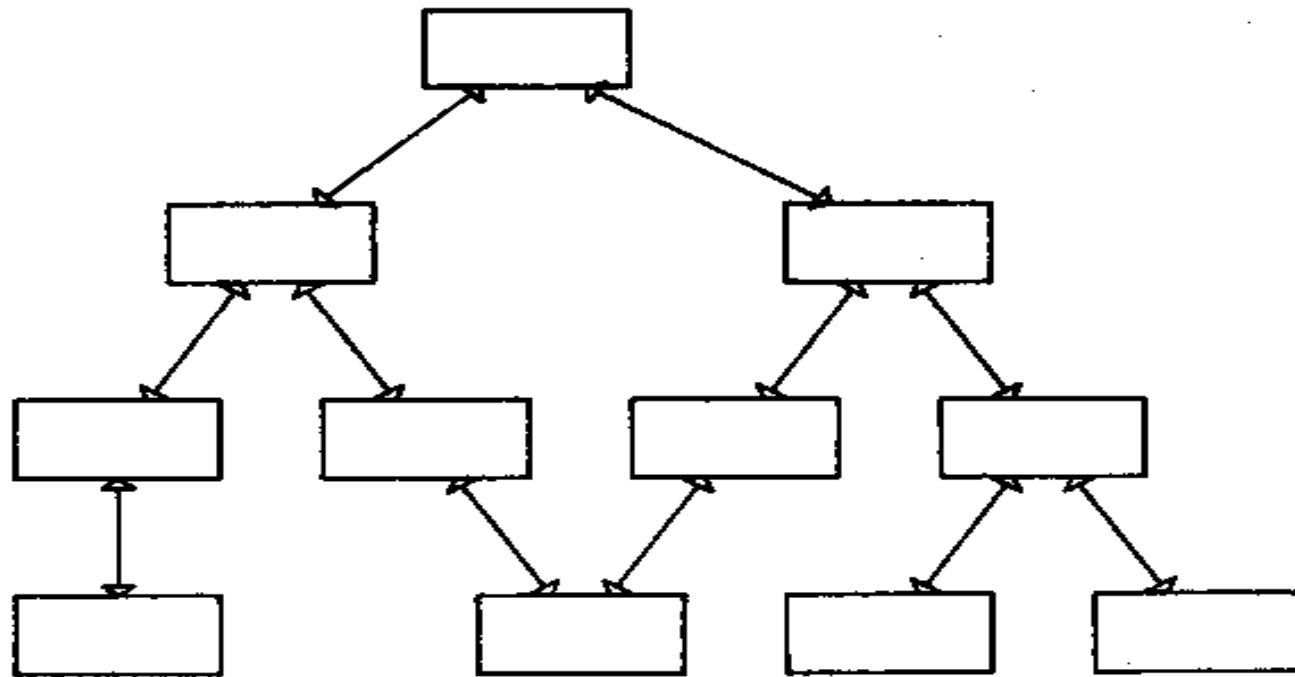


Fig. 13. Functional Decomposition, Structured Design

- **Grid charting:** This is a frequently used checking technique in several methods. Matrices confronting two aspects of the design, for instance organisational activities and data, are used to check for correctness and completeness, as well as for two dimensional sorting purposes of combinations of aspects of a design. Sometimes four aspects in one chart are used. See Figure 14. It was first described in Hartman et al. (1968). Methods like BSP (Business Systems Planning) (IBM, 1984) use grid charting extensively.

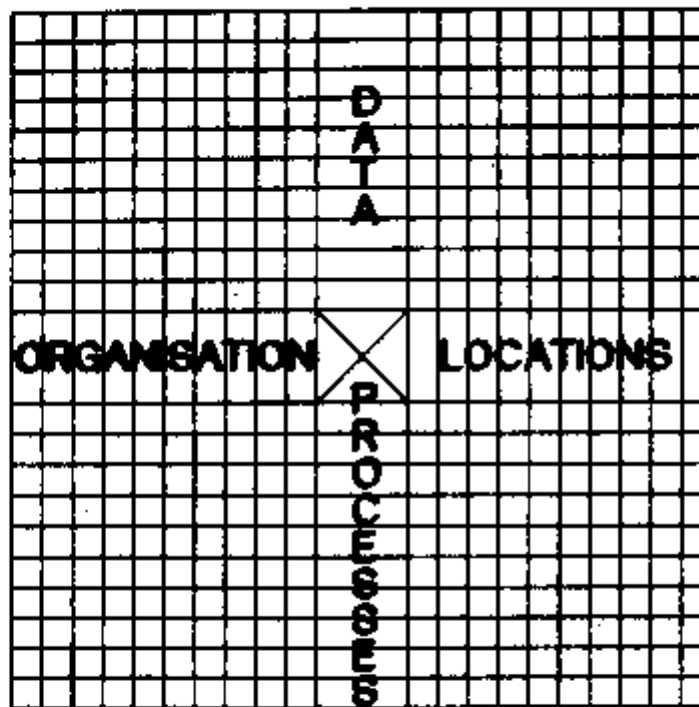


Fig. 14. Grid Charting, Four Aspects

- **Data synthesis:** This technique, used for the data model, synthesises the different users' views on the data in an information system to one single model, called the conceptual model. See Figure 15. It is the technique of integrating data by defining the relationships between the data. Vetter (1977) is an example of data synthesis.

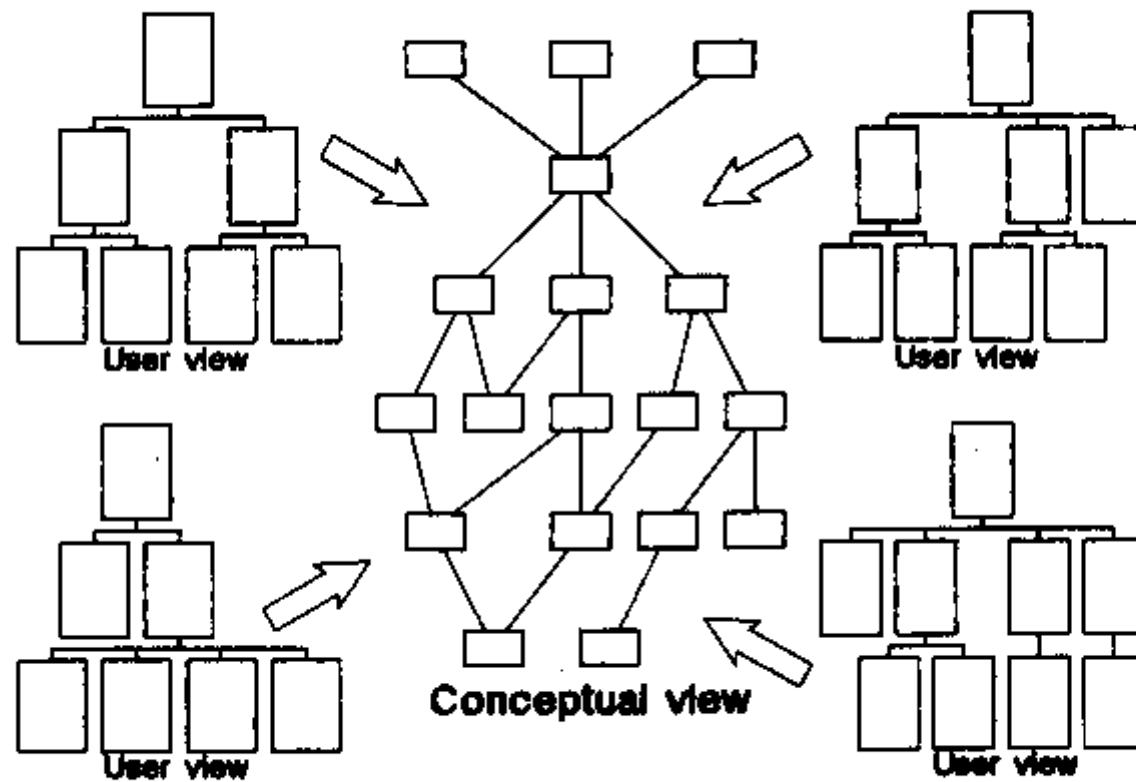


Fig. 15. Data Synthesis

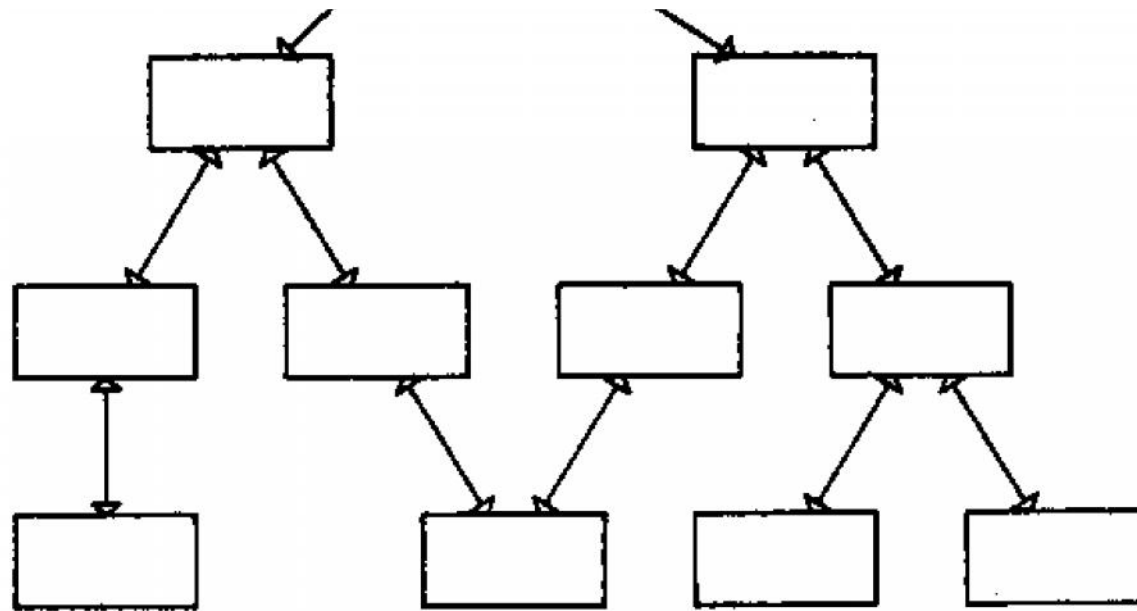


Fig. 13. Functional Decomposition, Structured Design

- **Grid charting:** This is a frequently used checking technique in several methods. Matrices confronting two aspects of the design, for instance organisational activities and data, are used to check for correctness and completeness, as well as for two dimensional sorting purposes of combinations of aspects of a design. Sometimes four aspects in one chart are used. See Figure 14. It was first described in Hartman et al. (1968). Methods like BSP (Business Systems Planning) (IBM, 1984) use grid charting extensively.

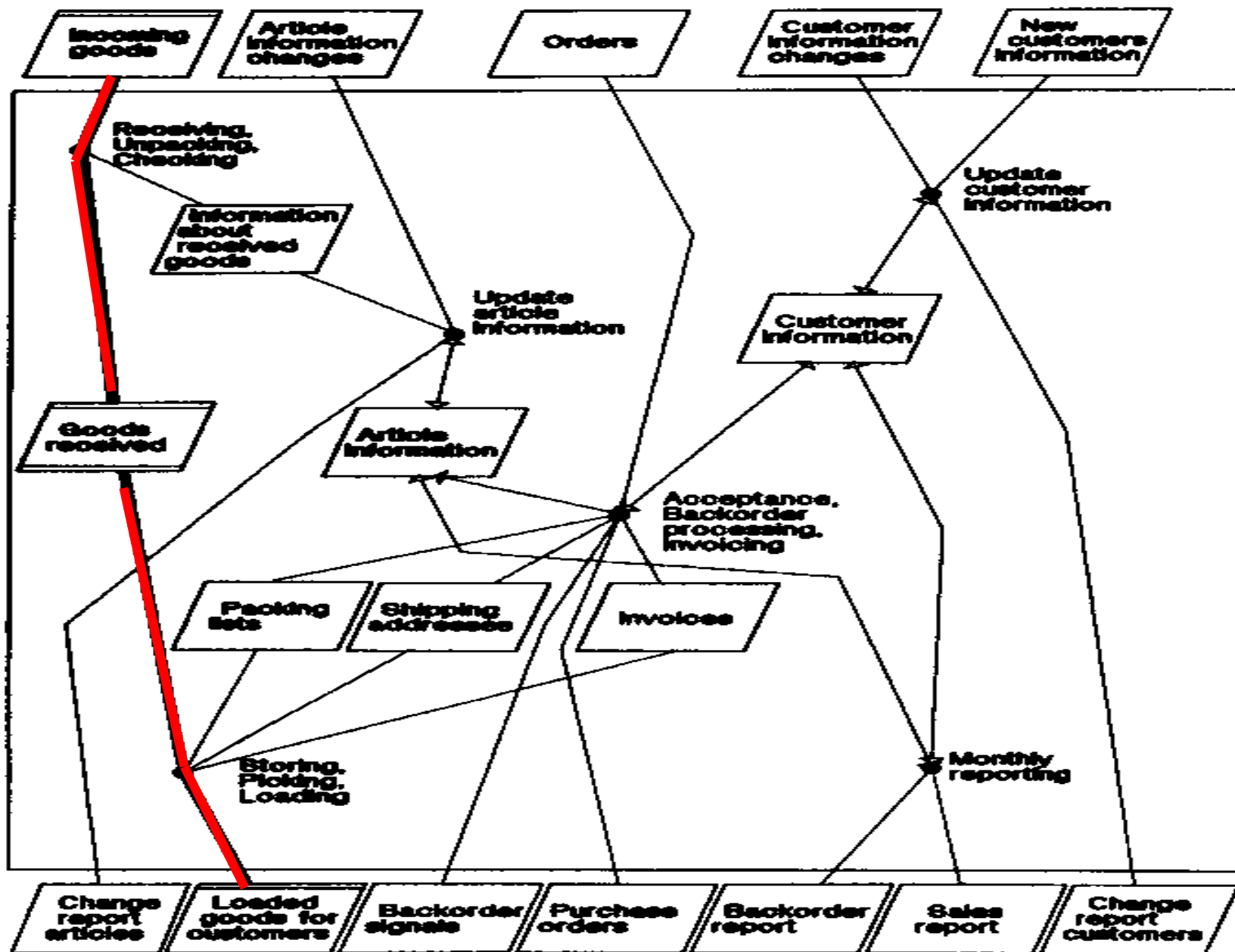


Fig. 16. Precedence Graph

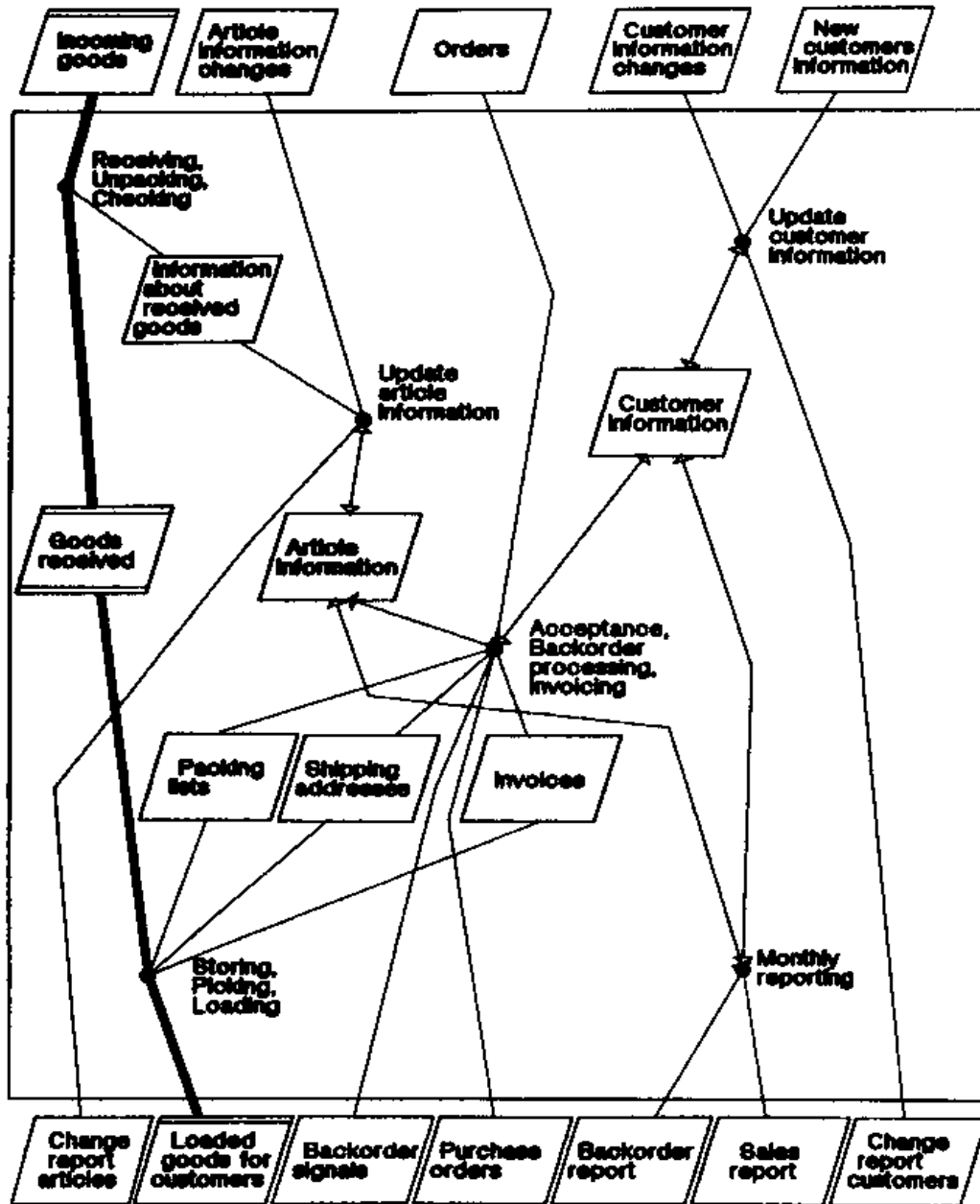


Fig. 16. Precedence Graph

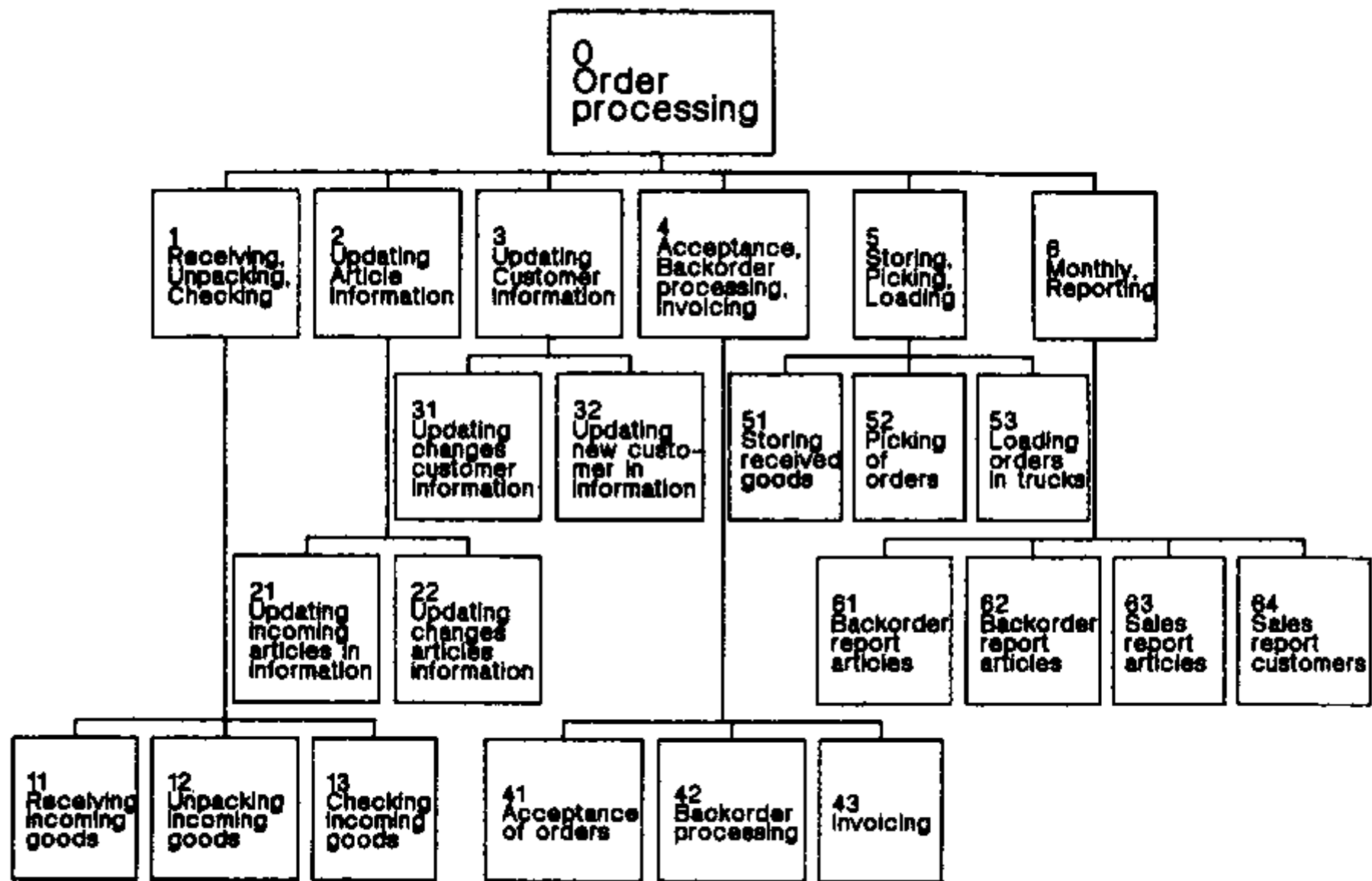


Fig. 17. Hierarchy Chart

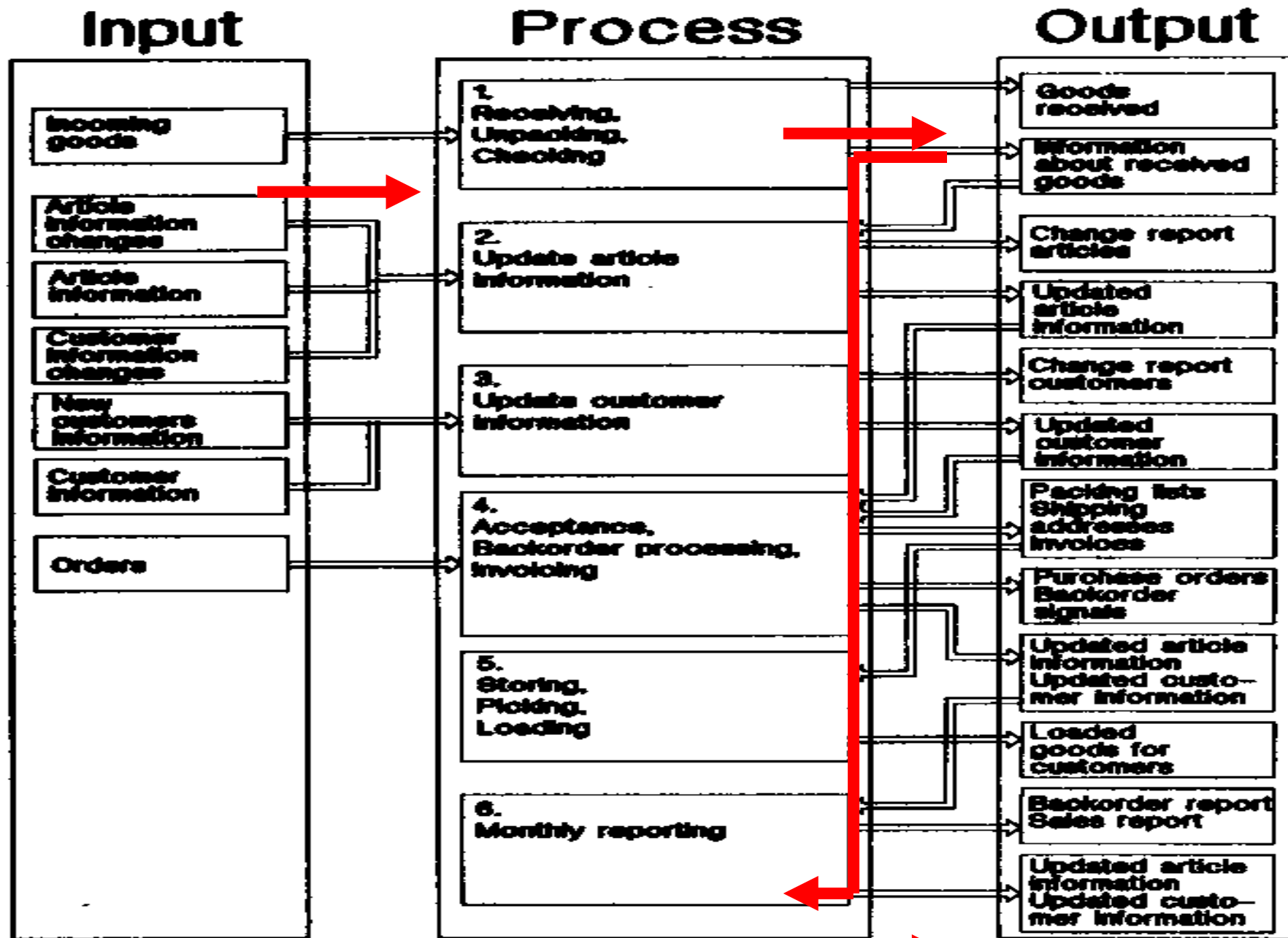


Fig. 18. Input — Process — Output Chart



Fig. 18. Input — Process — Output Chart

- **What is in the boxes and what is in the lines and knots between the lines seems to be inverted: compare precedence graphs and data flow diagrams (see also Figure 53 and Figure 54). Some say it does not matter what is in the box and what is in the lines and knots. Our observation is that the preferred notation of a technique requires what is decomposed to be in a box and the derived items to be in the lines and knots. So: precedence analysis decomposes the inputs and outputs which are expressed in the boxes (parallelograms), func-**

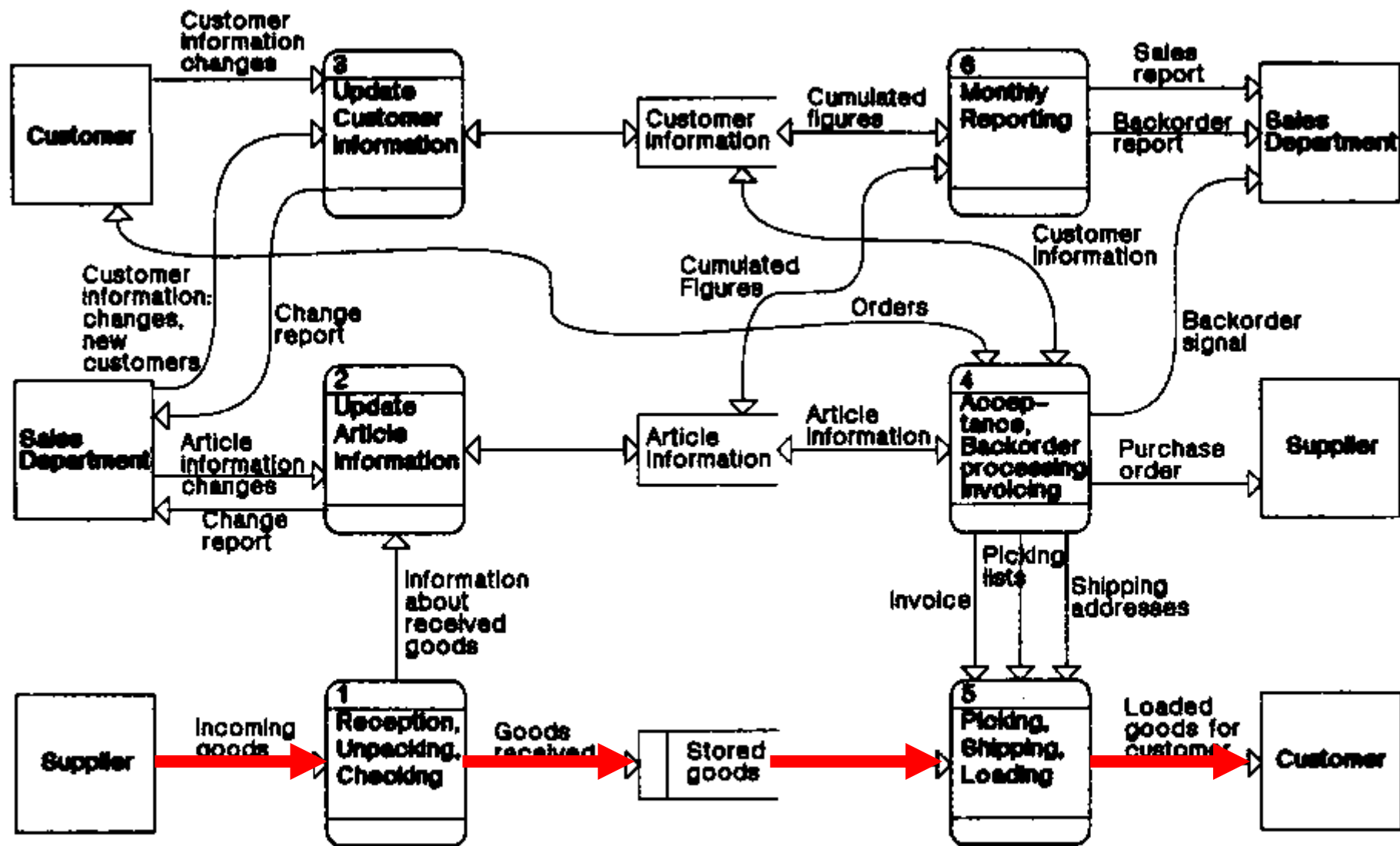


Fig. 19. Data Flow Diagram

1.	Deliver the Backorder from Minimum stock?	N	N	Y	Y	Y	Y
2.	Stock - Minimum stock >= Quantity to deliver?	N	Y	N	N	N	Y
3.	Stock >= Backorder quantity to deliver from Minimum stock?	-	-	N	N	Y	-
4.	Stock = 0?	-	-	N	Y	-	-
1.	Backorder quantity = Backorder quantity	X	-	-	X	-	-
2.	Quantity to deliver = Backorder quantity	-	X	-	-	-	X
3.	Stock = Stock - Backorder quantity	-	X	-	-	-	X
4.	Delete Backorder (completely shipped)	-	X	-	-	-	X
5.	Quantity to deliver = Backorder quantity to deliver from Minimum stock - Stock	-	-	X	-	-	-
6.	Stock = 0	-	-	X	-	-	-
7.	Give Purchase signal	-	-	X	-	-	-
8.	Backorder quantity = Backorder quantity - Quantity to deliver	-	-	X	-	X	-
9.	Quantity to deliver = Backorder quantity to deliver from Minimum stock	-	-	-	-	X	-
10.	Stock = Stock - Quantity to deliver	-	-	-	-	X	-

Fig. 20. Decision Table

Deliver Backorder

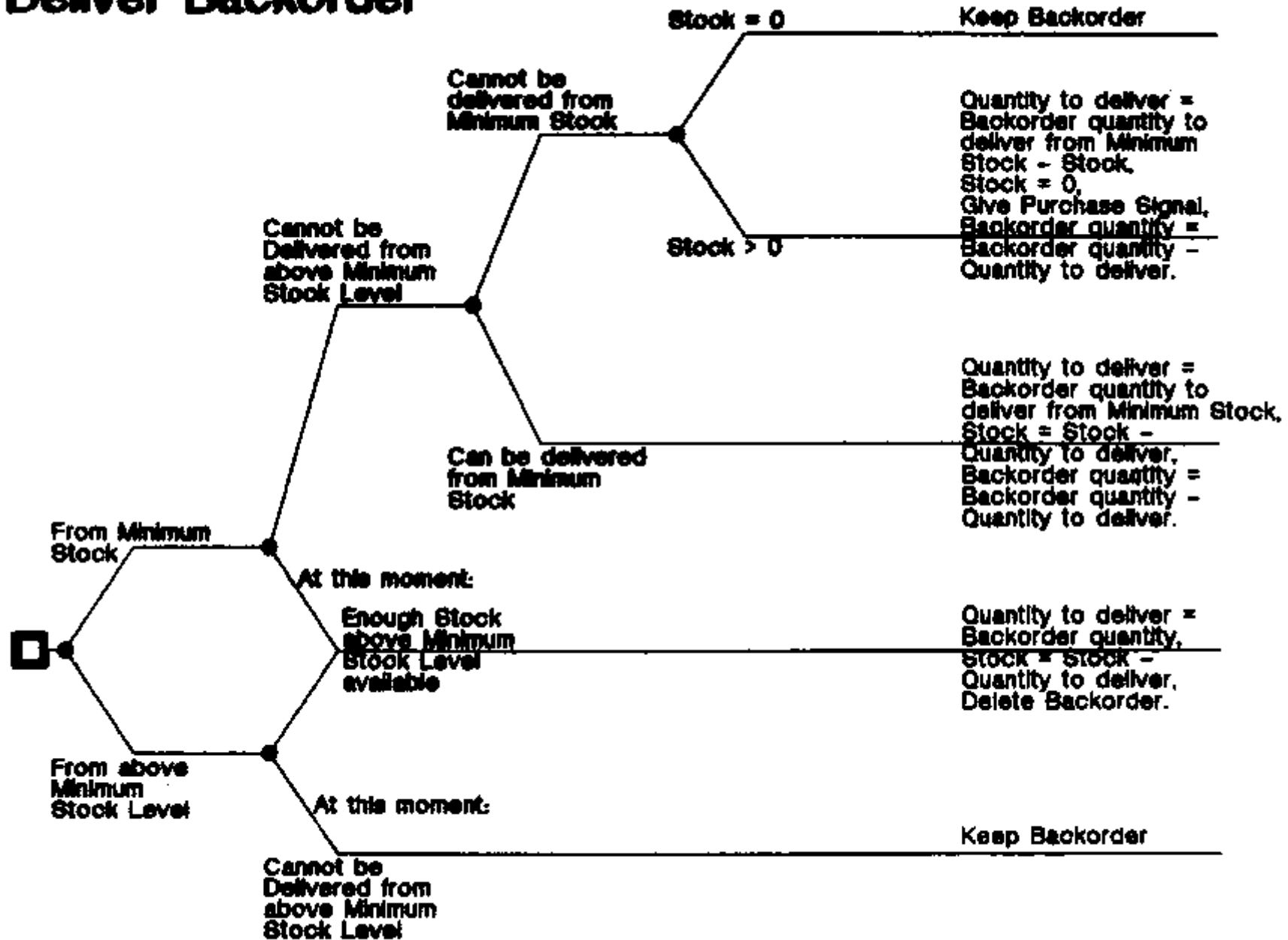


Fig. 21. Decision Tree



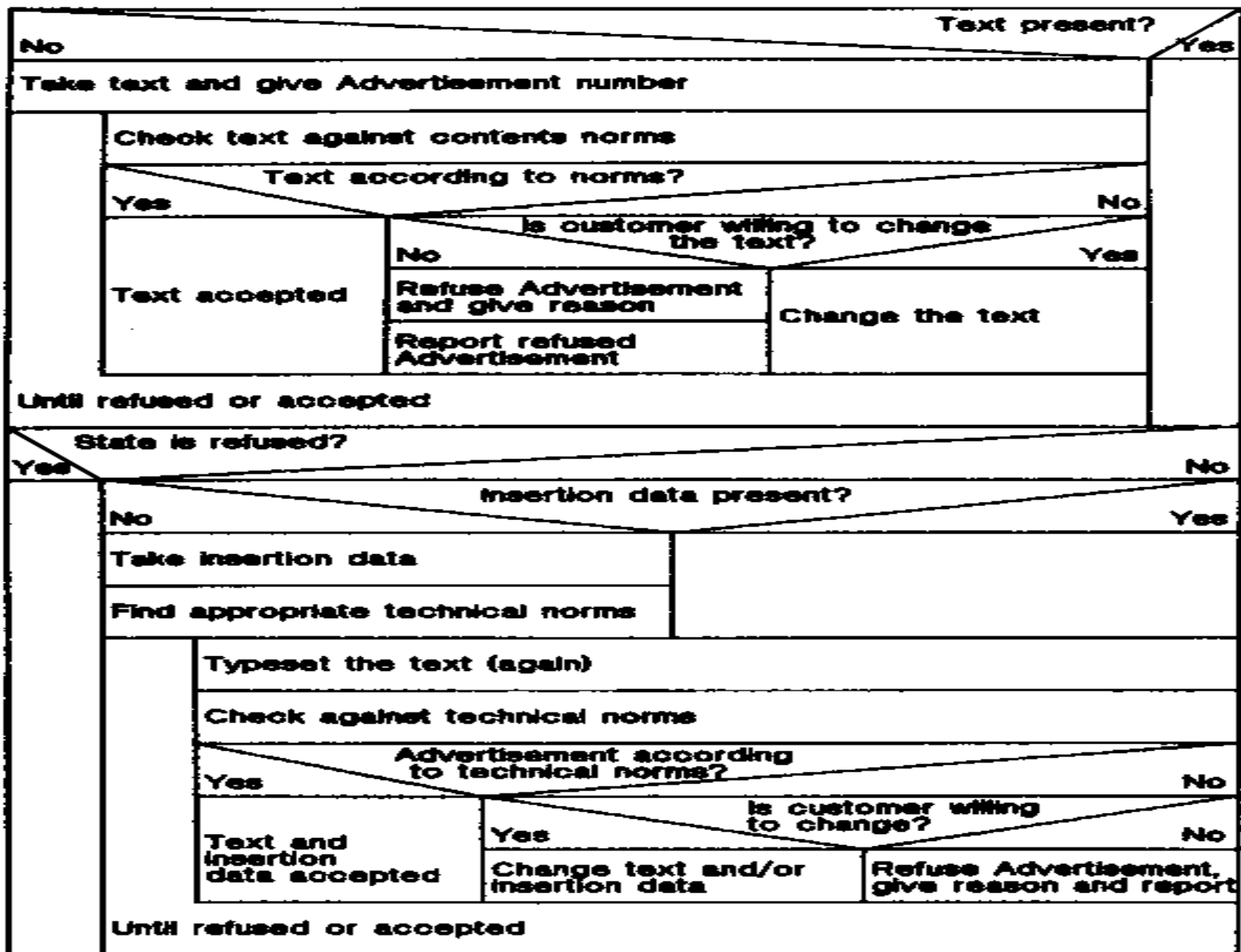


Fig. 22. Nassi-Shneiderman Diagram



```
BEGIN Take Ad.-text
IF (Advertisement text not present) THEN
  Take text and give Ad.-number
  DO UNTIL (Text refused or accepted)
    Check text against content norms
    IF (Text is according to the norms) THEN
      Text is accepted
    ELSE
      IF (Customer is willing to change text) THEN
        Change the text
      ELSE
        Refuse Ad. and give reason to customer
        Report refused Ad.
      ENDIF
    ENDIF
  ENDDO
ENDIF
IF (State is not refused) THEN
  IF (insertion data is not present) THEN
    Take insertion data
    Find appropriate technical norms
  ENDIF
  DO UNTIL (Technique refused or accepted)
    Typeset the Ad. (again)
    Check against technical norms
    IF (Ad. is according to technical norms) THEN
      Text and insertion data accepted
    ELSE
      IF (Customer is willing to change) THEN
        Change text and/or insertion data
      ELSE
        Refuse Ad. and give reason to customer
        Report refused Ad.
      ENDIF
    ENDIF
  ENDDO
ENDIF
END Take Ad.-text
```

Fig. 23. Structured English, Pseudo Code

Matematikai kitérő –

