Chapter 10 Computer Architectures for Urban Planning

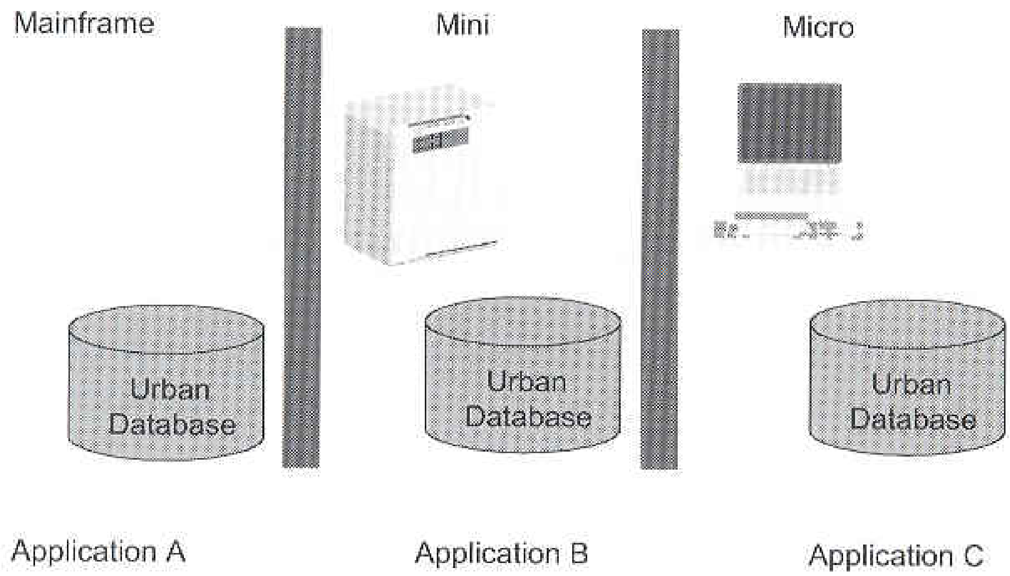

Figure 10.1 Separation between several urban databases

10.1 Generalities about info sys architecture

- several kinds of architectures for info sys are possible

  : centralized, decentralized, client-server, cooperative ..

<u>Centralized DBs</u>

- centralization means the separation of different applications

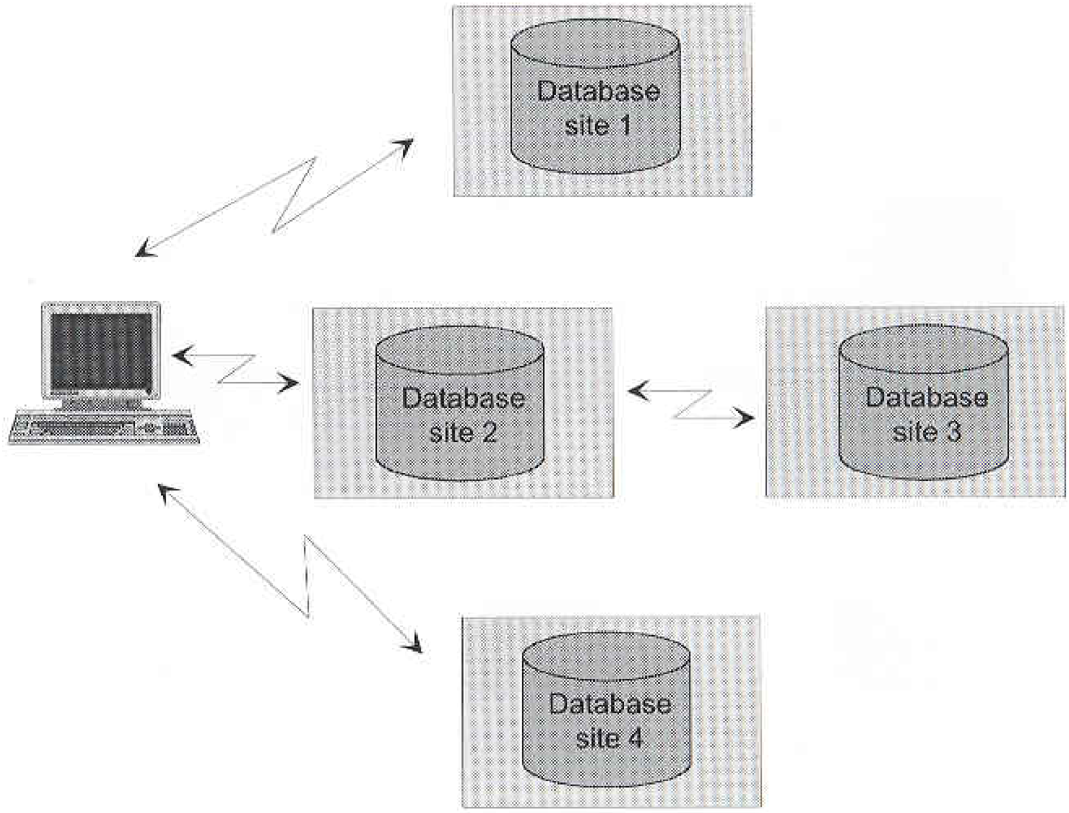

Figure 10.2 A single workstation or PC can access several remote database systems which may be located at different sites.

<u>Multi database systems</u>

- connect independent computers by networks

  users can develop / run tasks on several machines

  communication between machines is headache -> interoperability problem

<u>Cooperative info sys</u>

- sys enables several people & machines work together

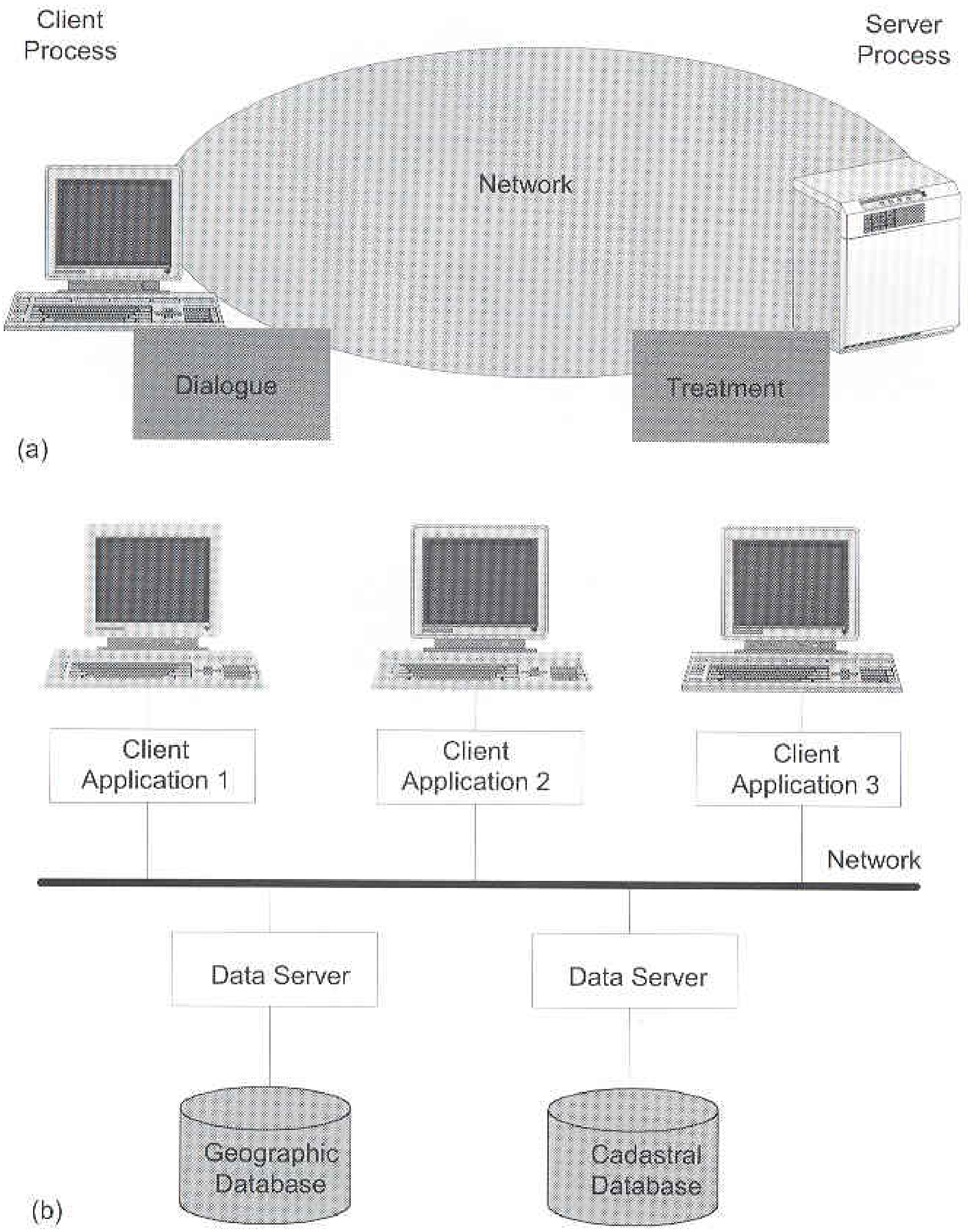

Figure 10.3 Principles of the client-server architecture. (a) The client process and the server process. (b) A small example in a planning context.

10.2 Client-server architecture

- client & server are independent but communicating

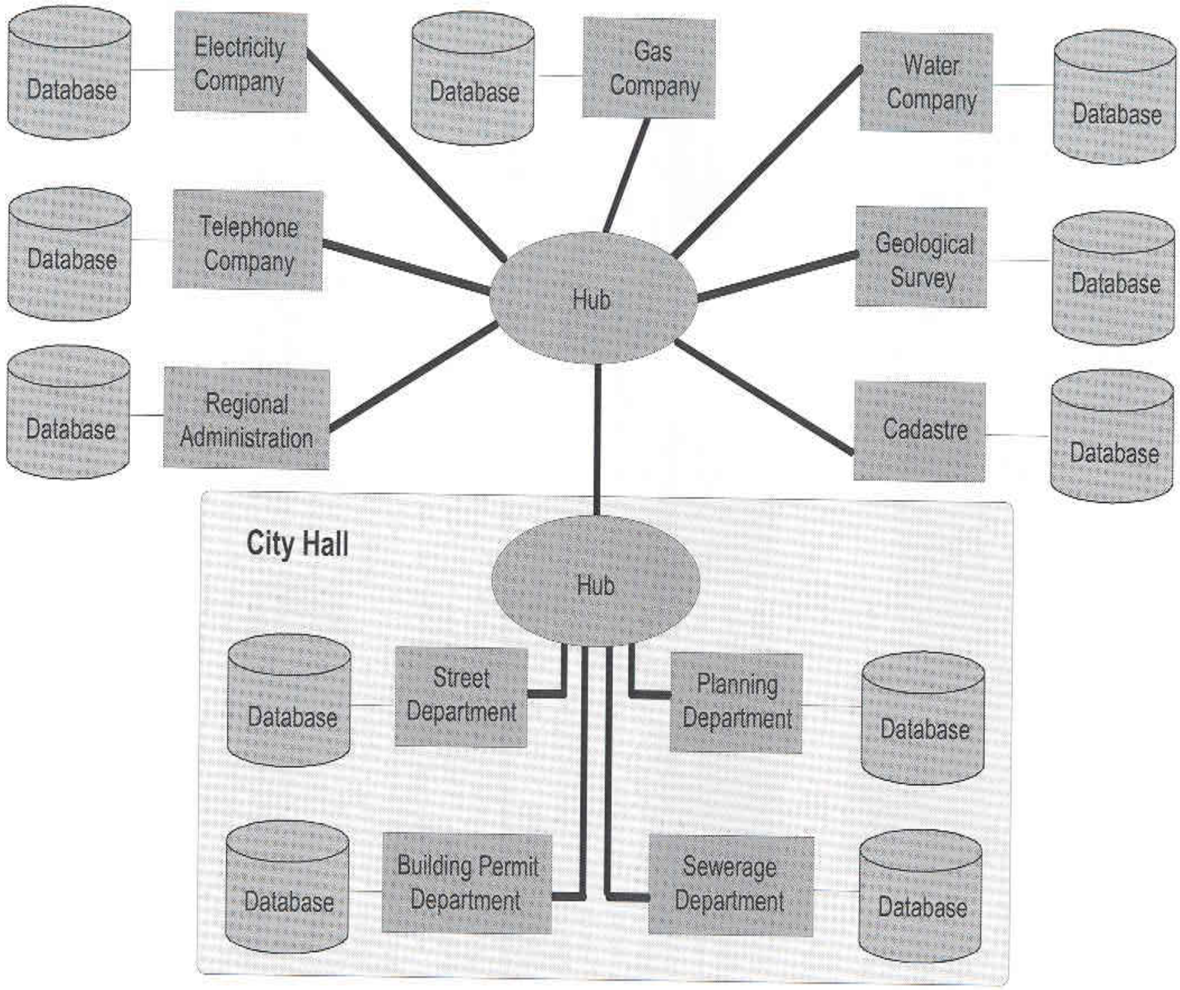

Figure 10.4 Example of a client-server architecture within a local authority.

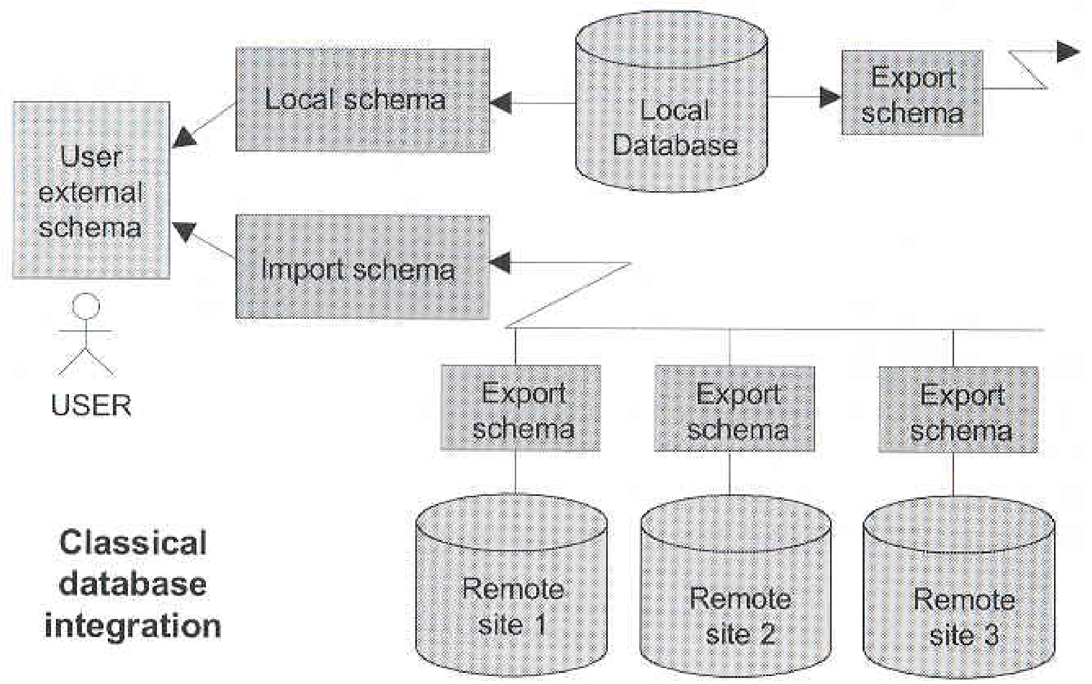

Figure 10.5 Relationships between schemata in multidatabases.

10.3 Federating several database systems

<u>Relationships between schemata in multi DBs</u>

- schema : structure of a DB
- relations between schemata

  : external schema = union of local schema & import schema

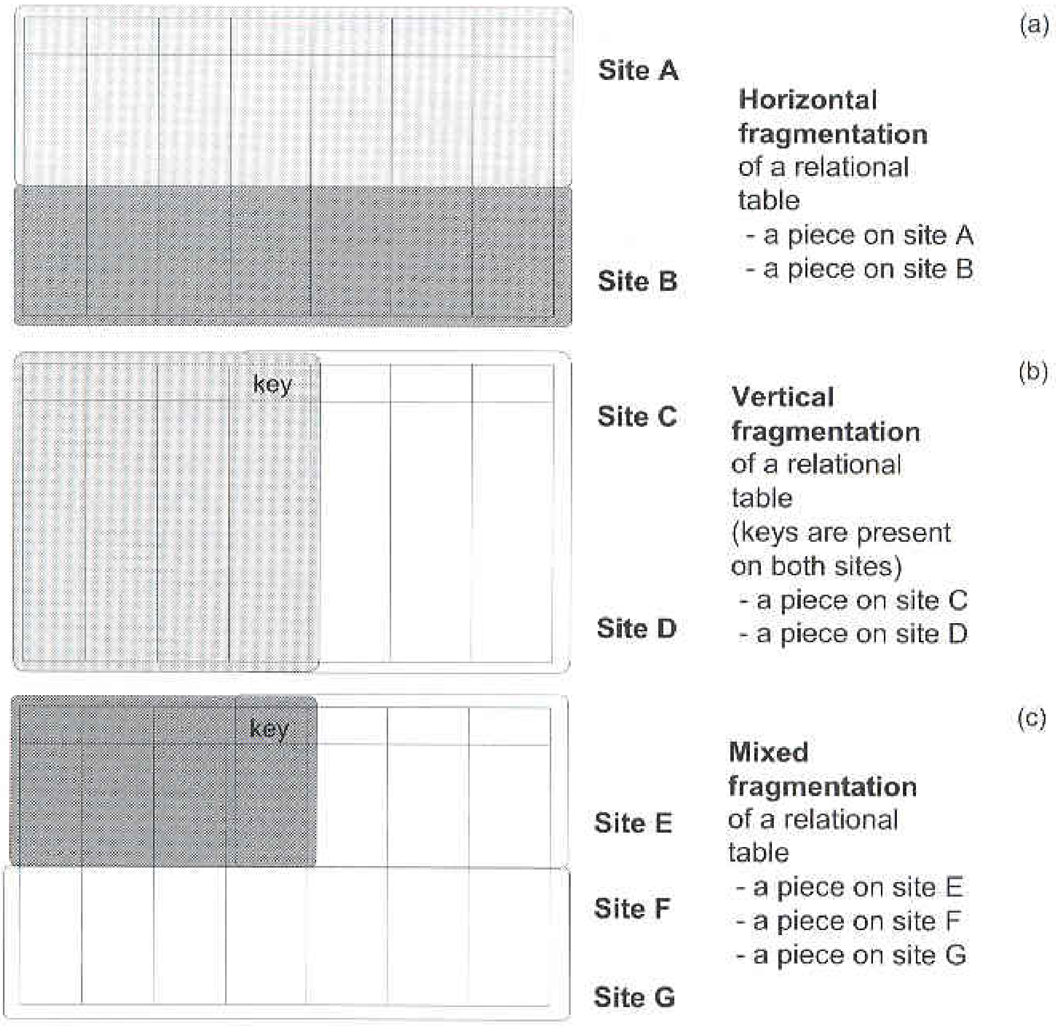

Figure 10.6 Horizontal, vertical and mixed fragmentation in relational databases.

<u>Fragmentation in relation DBs</u>

- horizontal, vertical, mixed fragmentation

  horizontal : e.g. a relation CUSTOMER can be split according to marketing sectors

  vertical : accounting info in a site & delivery info in another site

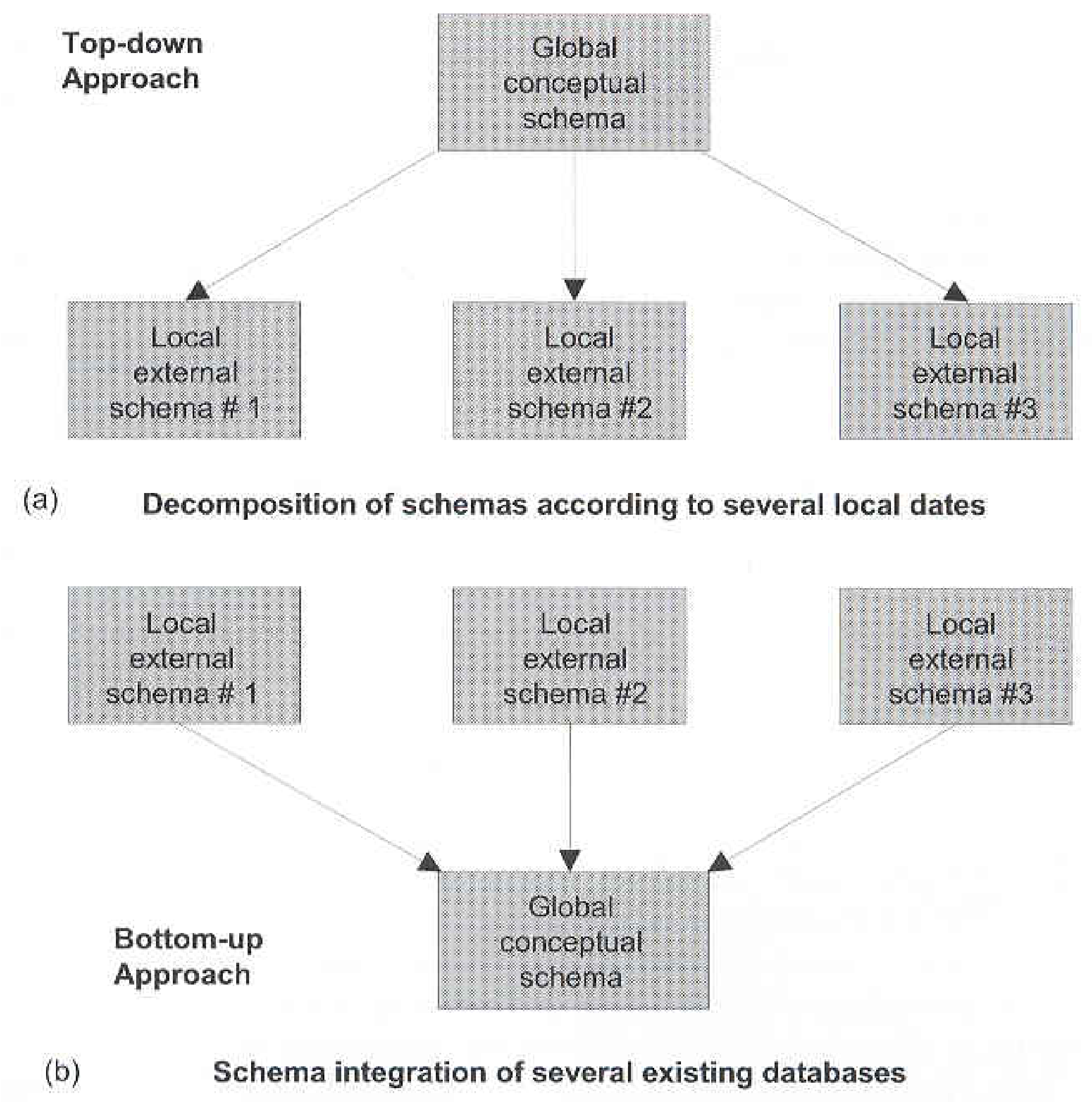

Figure 10.7 The top-down and the bottom-up schema designs. (a) Top-down or global-to-local approach for decomposing databases. (b) Bottom-up or local-to-global approach for integration.

<u>Designing multi DBs</u>

- two main cases

  build a distributed system from scratch (top-down)

  federate different existing DBs (bottom-up)

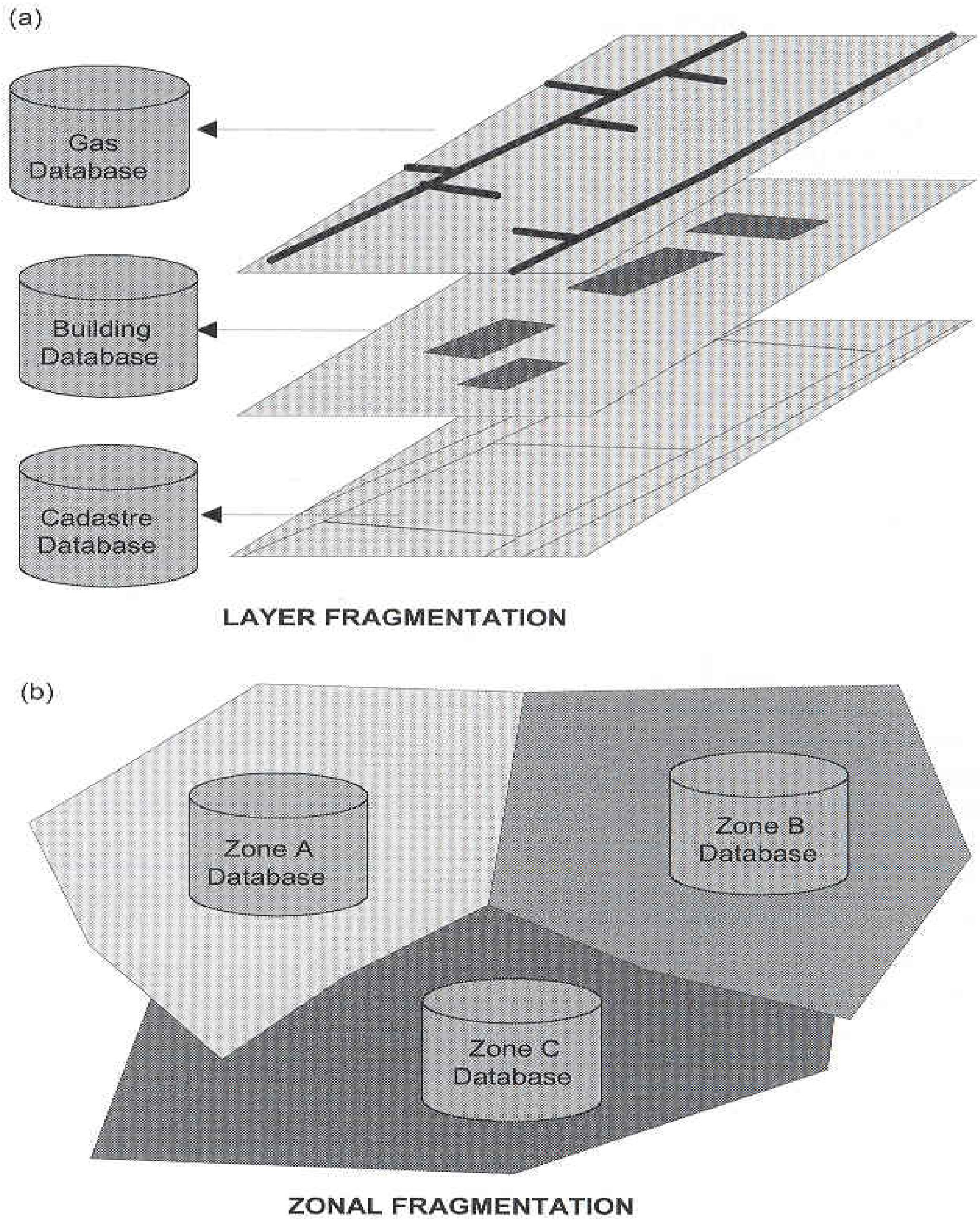

Figure 10.8 Zonal and layer fragmentation in distributed geographic databases. (a) Zonal or geographic partitioning. (b) Layer or thematic partitioning.

<u>Spatial & thematic fragmentation</u>

- spatial partitioning : zonal fragmentation
- thematic partitioning : layer fragmentation

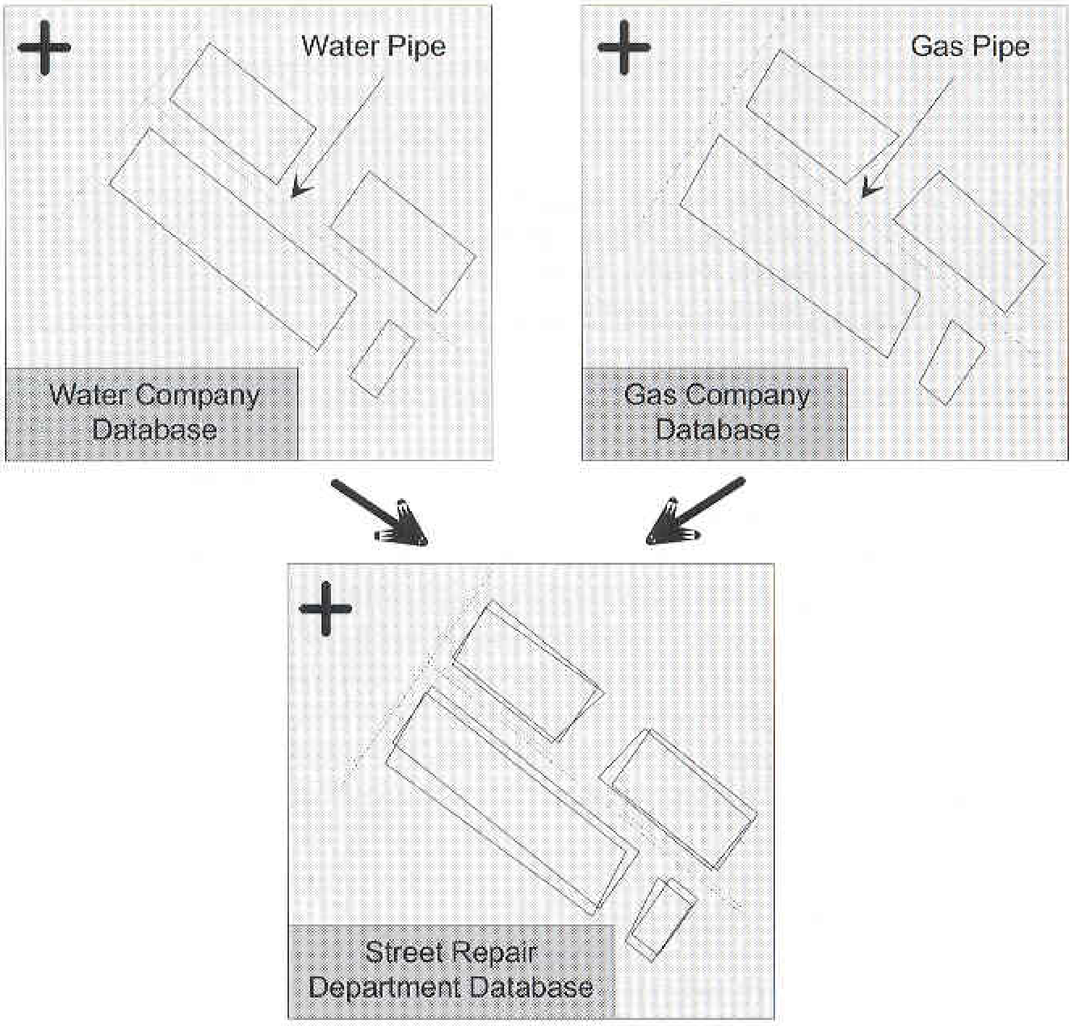

Figure 10.9 Example of geometric discrepancies in layer fragmentation.

Spatial schema integration

- one difficulty : geometric discrepancies

Gas Company Database (G-site)	
G-STREET	*(#street, street_name, (#axis_segment, width)*)*
G-SEGMENT	*(#segment, #point1, #point2)*
G-POINT	*(#point, x, y)*
G-PIPE	*(#edge, #node1, #node2)*
G-NODE	*(#node, x, y, z, type)*
Water Company Database (W-site)	
W-STREET	*(#street, (#right_segment, order)*, (#left_segment, order)*)*
W-SEGMENT	*(#segment, #from_point, #to_point)*
W-POINT	*(#point, x, y)*
W-PIPE (#edge, #from_node, #to_node)	
W-NODE	*(#node, x, y, z, (#edge)*, category)*
Street Repair Department Database (SR-site)	
SR-STREET	*(#street, street_name, (#parcel_segment)*,(kerb_segment)*)*
SR-SEGMENT	*(#segment, #point1, #point2, begin_address, end_address)*
SR--POINT	*(#point, x, y)*
SR-G-PIPE	*(#edge, #node1, #node2)*
SR-G-NODE	*(#node, x, y, depth, type)*
SR-W-PIPE	*(#edge, #node1, #node2)*
SR-W-NODE	*(#node, x, y, depth, type)*

Figure 10.10 Example of semantic discrepancies in distributed urban databases.

- semantic discrepancies of different DBs

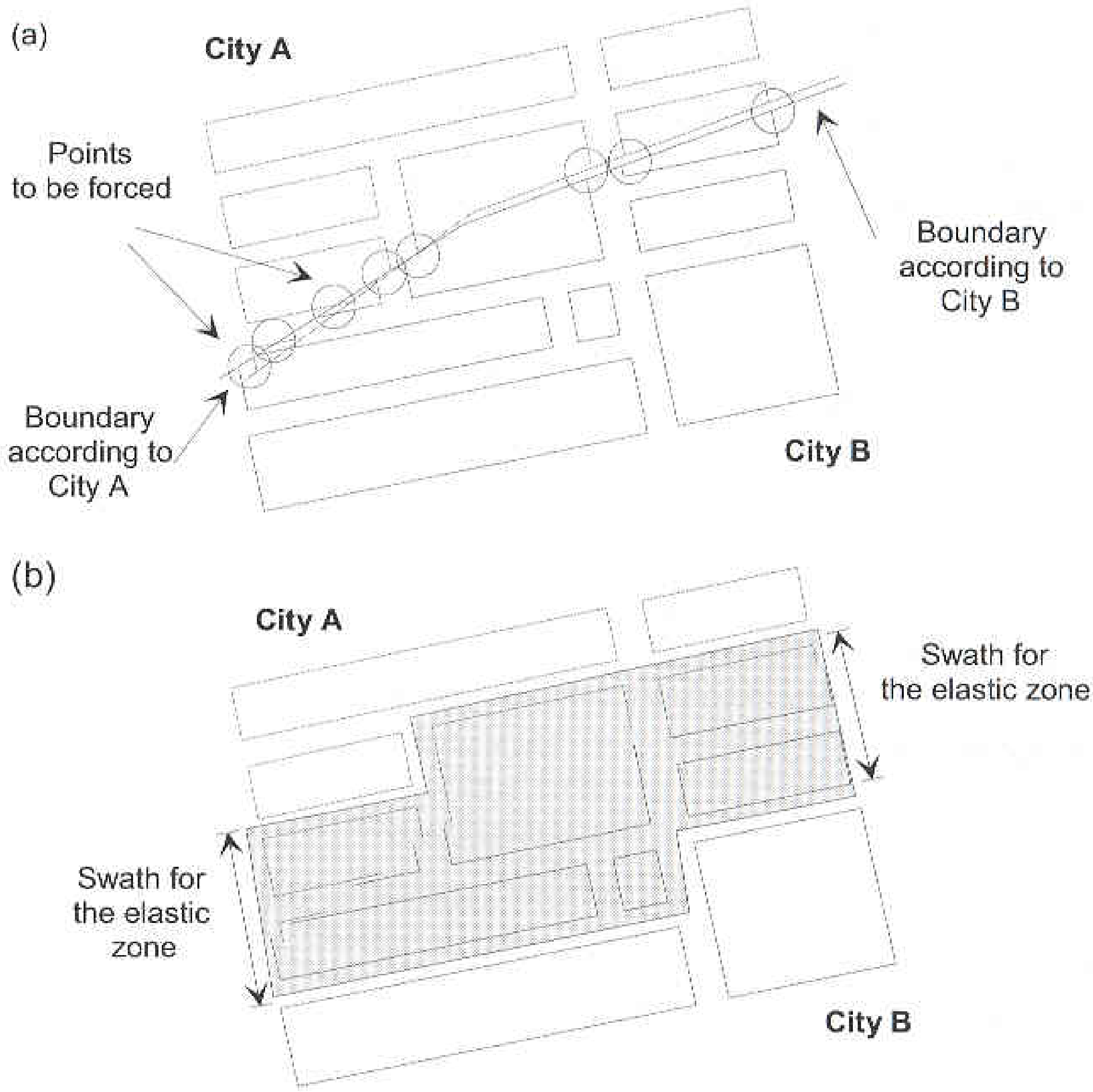

Figure 10.11 Elastic zone at the boundary.

- boundary problem due to measurement error
  : elastic transformation w/ constraints must be launched
  a swath is defined for the elastic zone

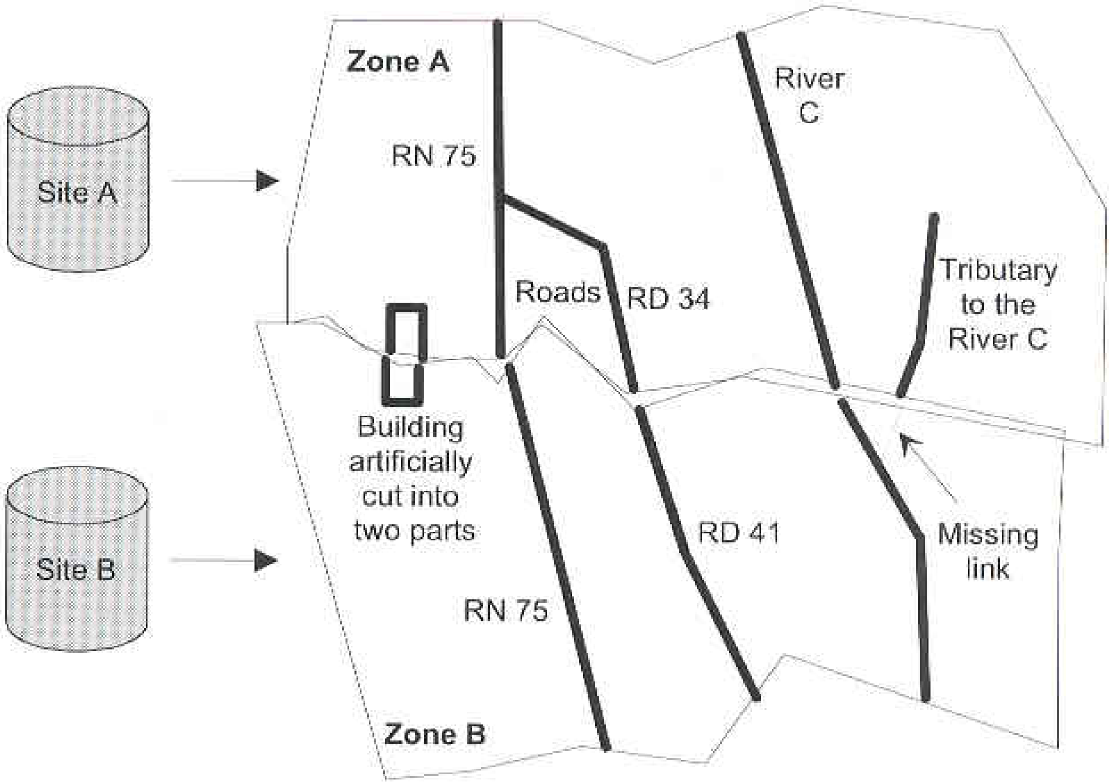

Figure 10.12 Examples of problems to solve in spatial multidatabase systems.

- Boundary alignment problem

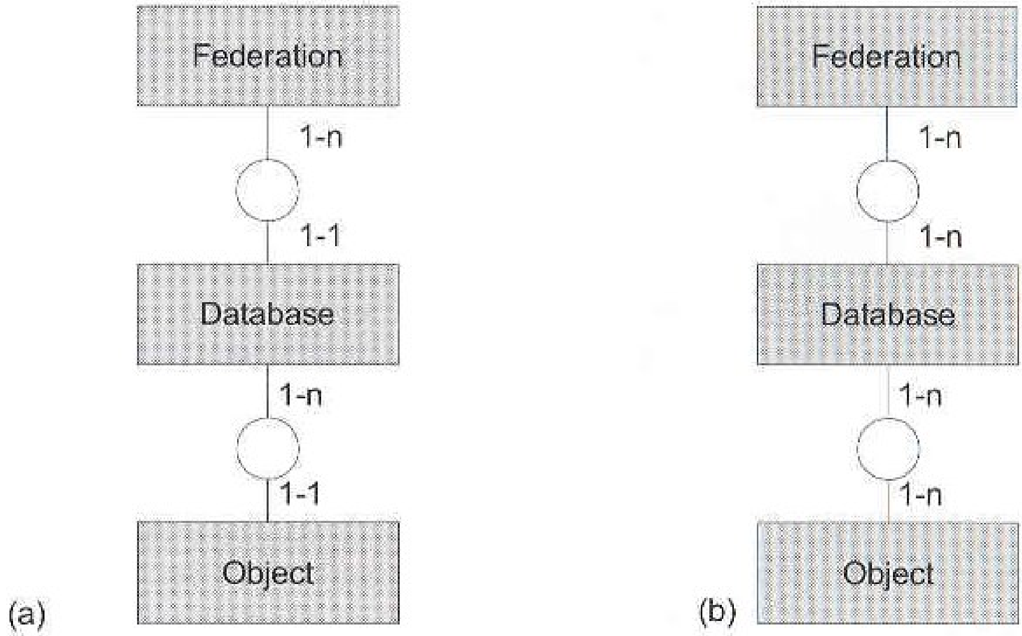

Figure 10.13 Structure of federations. (a) Totally hierarchical, which is the more common case. (b) A database can belong to several federations, and an object can belong to several pages

- Structure of federation

  a. a DB belongs to only one federation

  b. a DB belongs to several federations

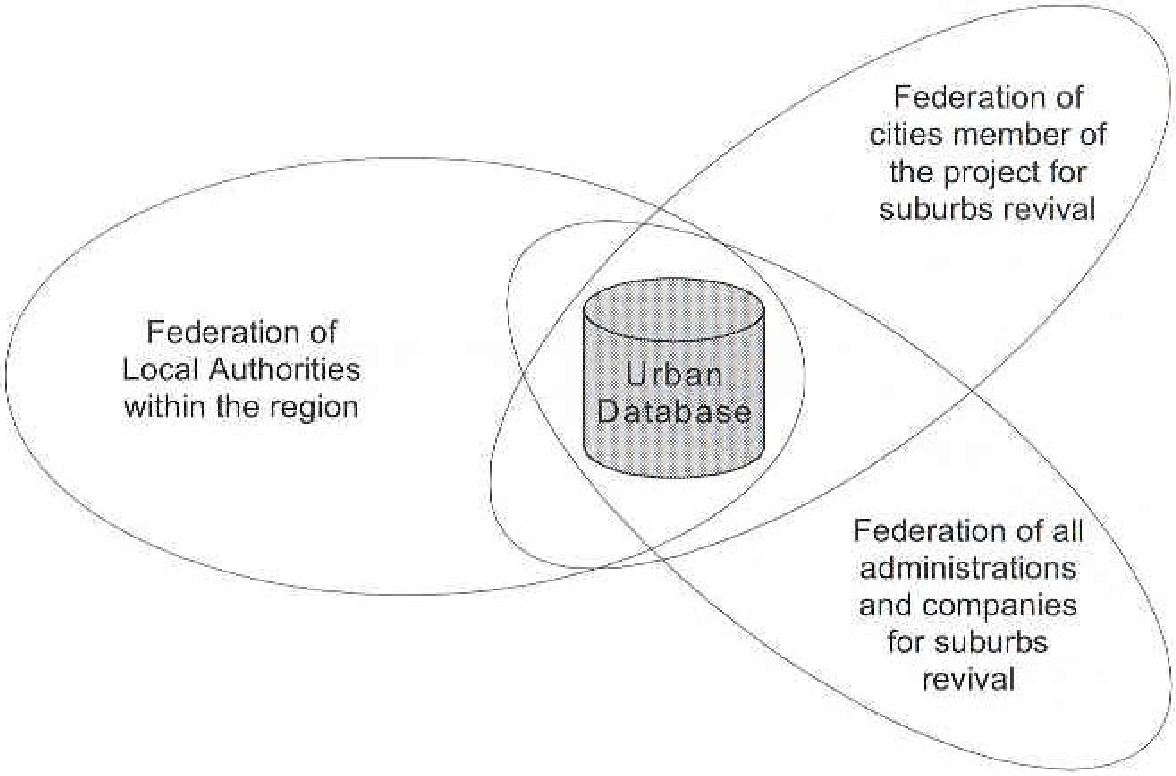

Figure 10.14 A city database can belong to several federations

- example: a city DB belongs to several federations

	Short term solutions	Medium term solutions	Long term solutions
Concepts for solutions	List of virtual tables (schemata)	Manual mediators	Ontology based
Connectability	Stovepipe connectivity	Interconnectivity	Interoperability
Organizational aspects	Application view	Departmental view	Organization view

Figure 10.15 The implementation continuum, from the application view to the global organisation view.

10.4 Interoperability in geoprecessing

- a sort of continuum can be found to solve the linking problem of several computer systems

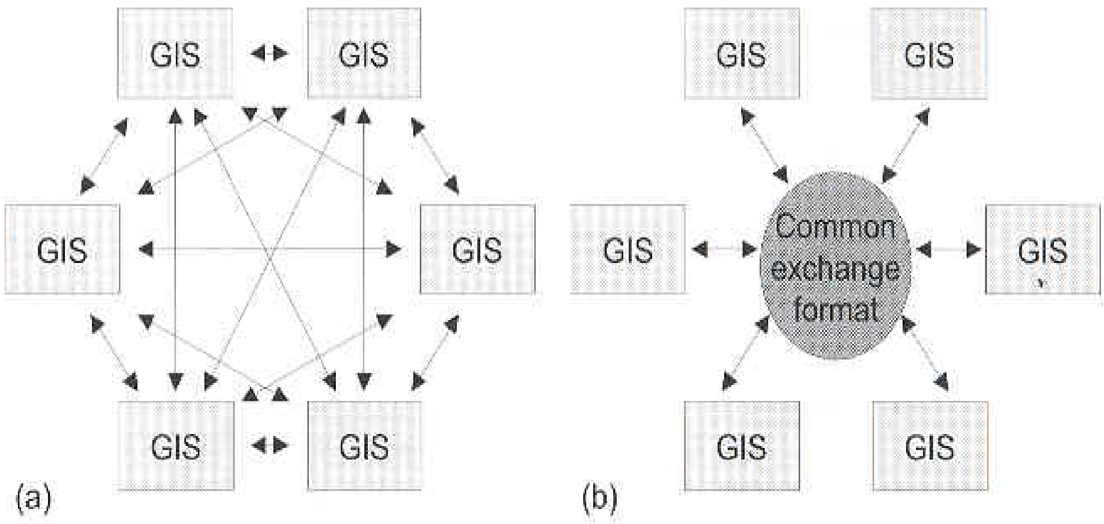

Figure 10.16 GIS interoperability. (a) Without a common exchange standard. (b) With a common exchange standard.

<u>Necessity of a common exchange format</u>

- if n different GISs, we need n×(n-1)/2 modules for adapting data
  -> common exchange format solves this problem
  -> Open GIS Consortium(OGC)'s OGIS
    * OGIS (Open Geodata Interoperability Specification)

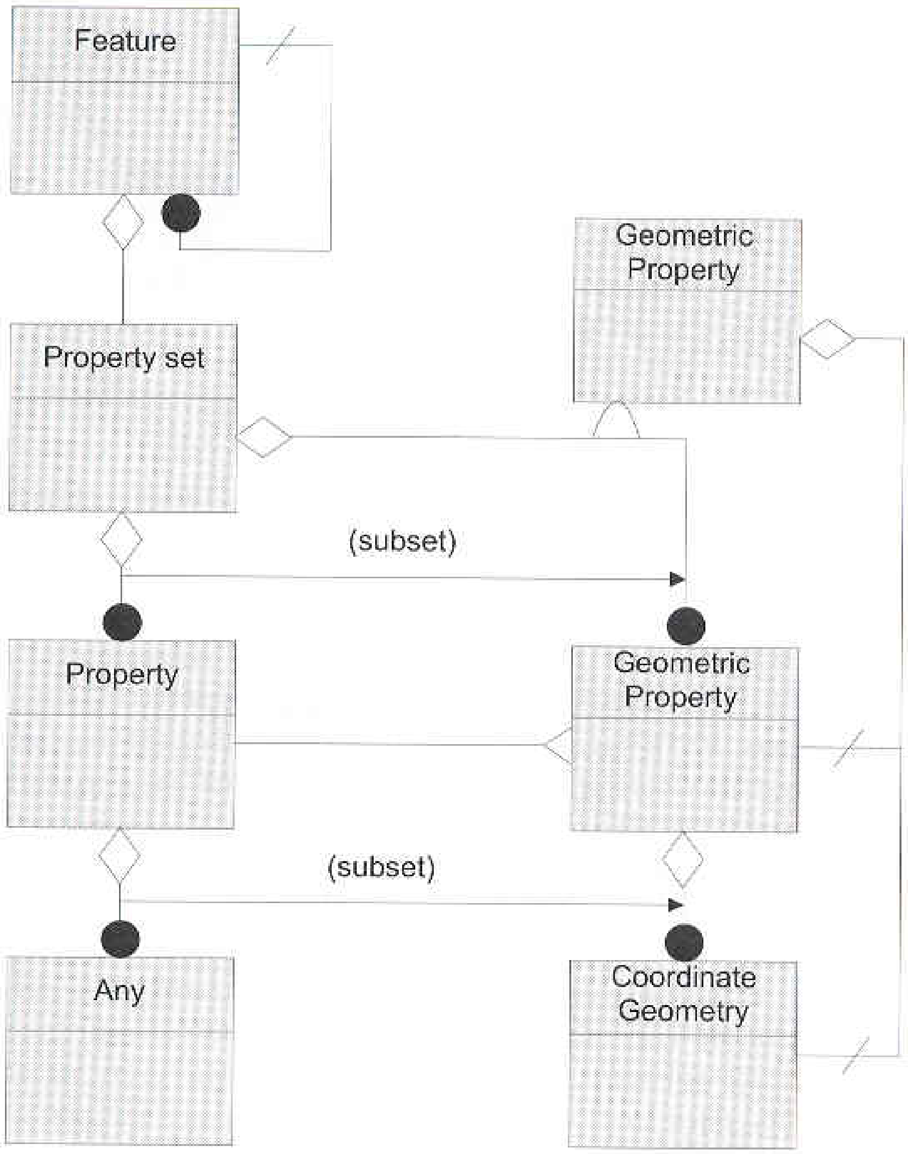

Figure 10.17 Essential model of OGIS feature type (Buelher and McKee 1996).

Introduction to OGIS

- two essential models : a feature & a coverage
- feature

  representation of a real world entity

  feature has a spatio-temporal location as attributes

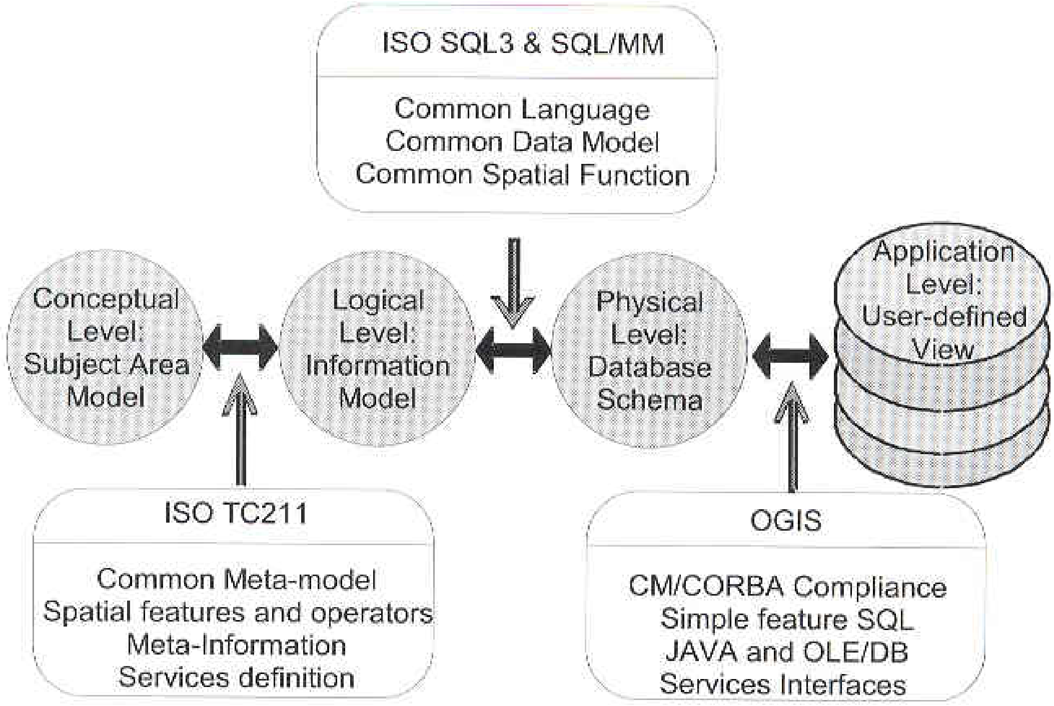

Figure 10.20 Linking standards in order to ensure interoperability (Kucera-O'Brien 1997, Evangelatos 1999).
Reprinted from Kucera, H.A. and Leighan, E. (1998) 'Mercator III: Toward a Canadian Geospatial Information Infrastructure', Technical Proposal Report to Natural Resources Canada, GeoAccess Division, March with kind permission of Henry Kucera.

Implementing interoperability

- one solution for implementing operability is to use mediators

  client - mediators – data server

- another solution is to base the architecture on existing standards

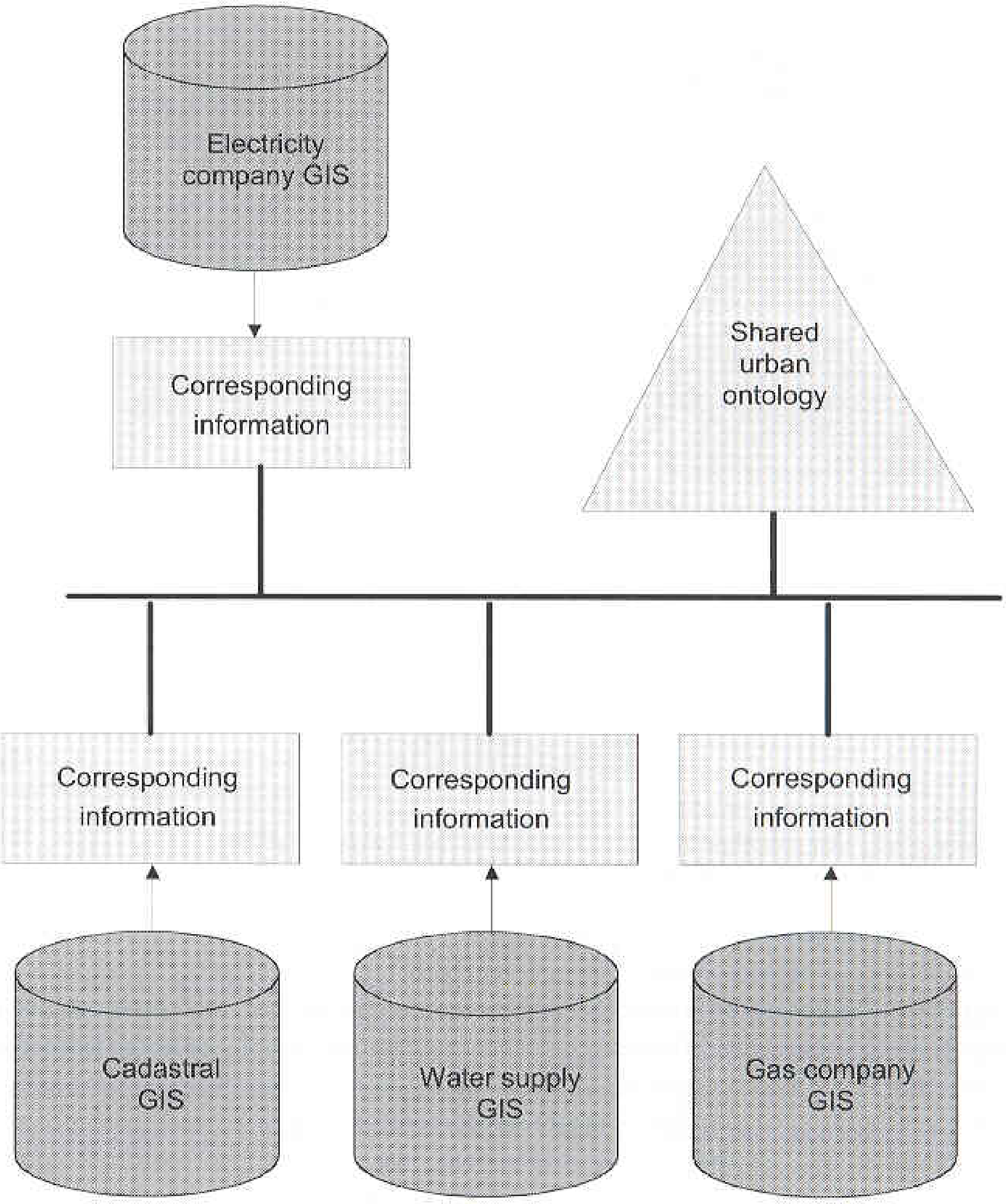

Figure 10.21 Architecture based on ontologies (Benslimane *et al.* 1999).

Ontology based interoperability

- ontology : a formalized vocabulary to describe data

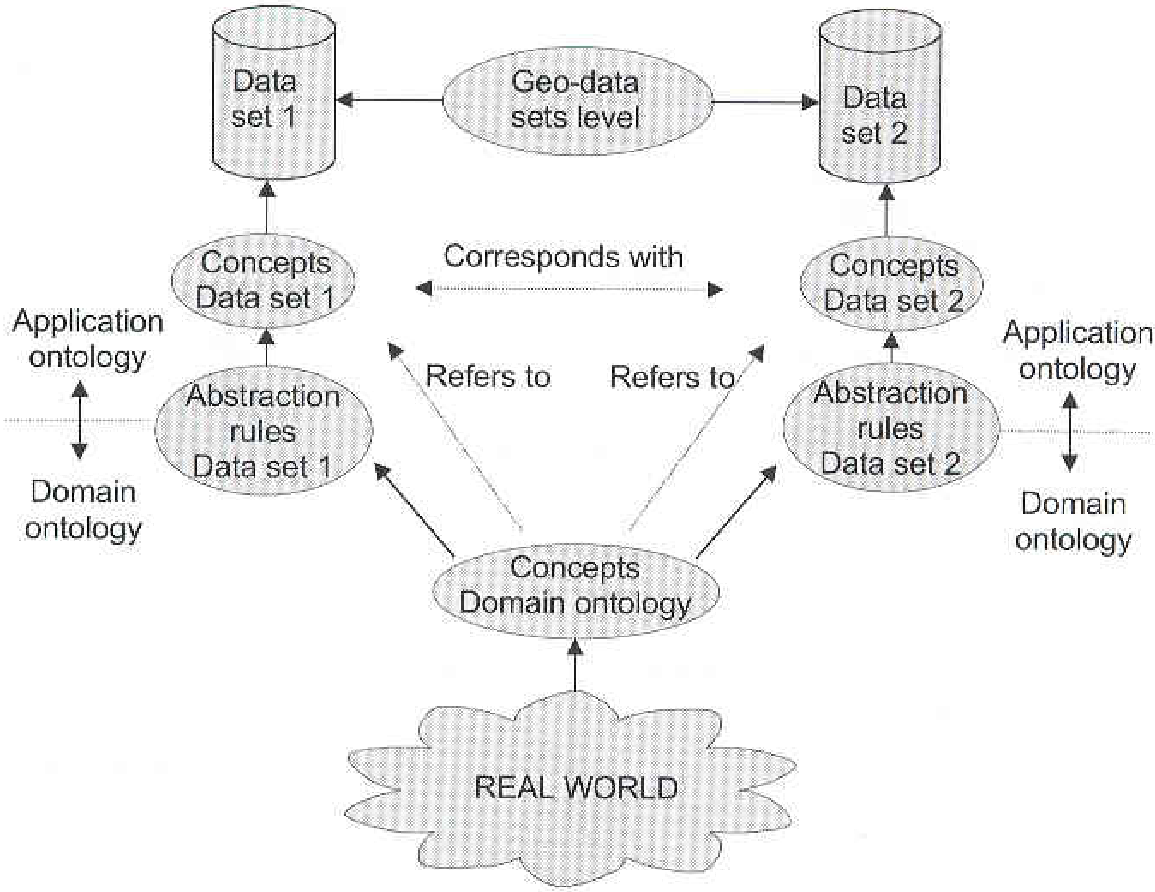

Figure 10.22 An example of using ontologies for interoperability.
From Uitermark *et al.* (1999).

- example

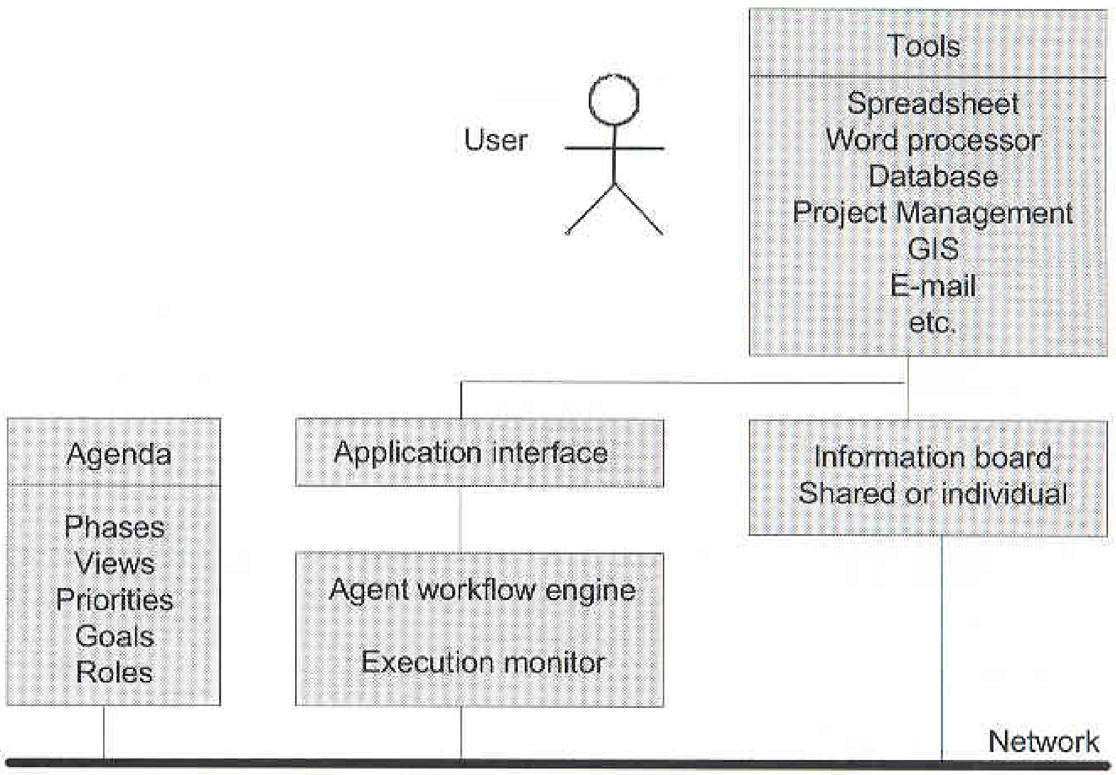

Figure 10.23 Architecture of a CSCW system.
After Mahling, Craven and Croft (1995).

10.5 Architecture for groupware

- CSCW tools can be implemented in various ways depend on types of coumputer systems

  : centralized approach, decentralized approach

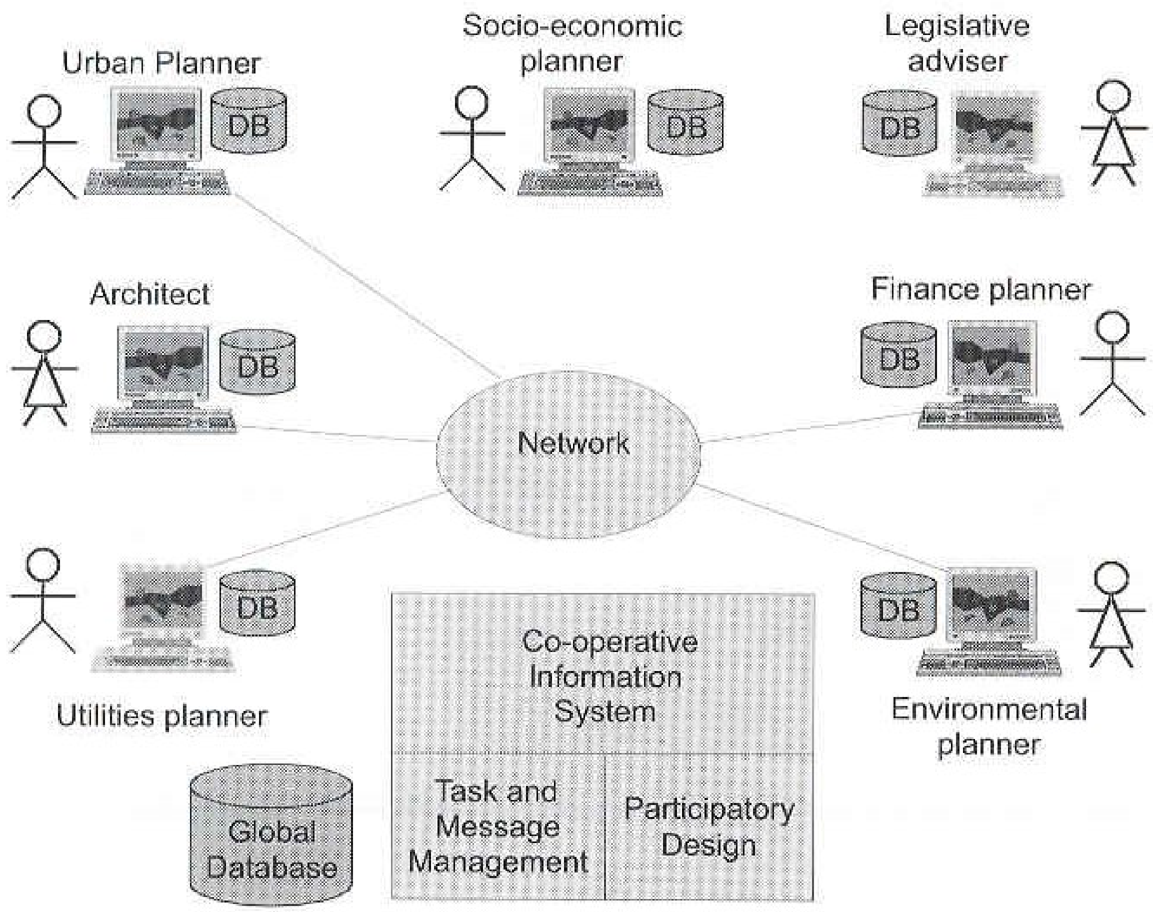

Figure 10.24 Architecture of a CSCW system for urban planning.

- example

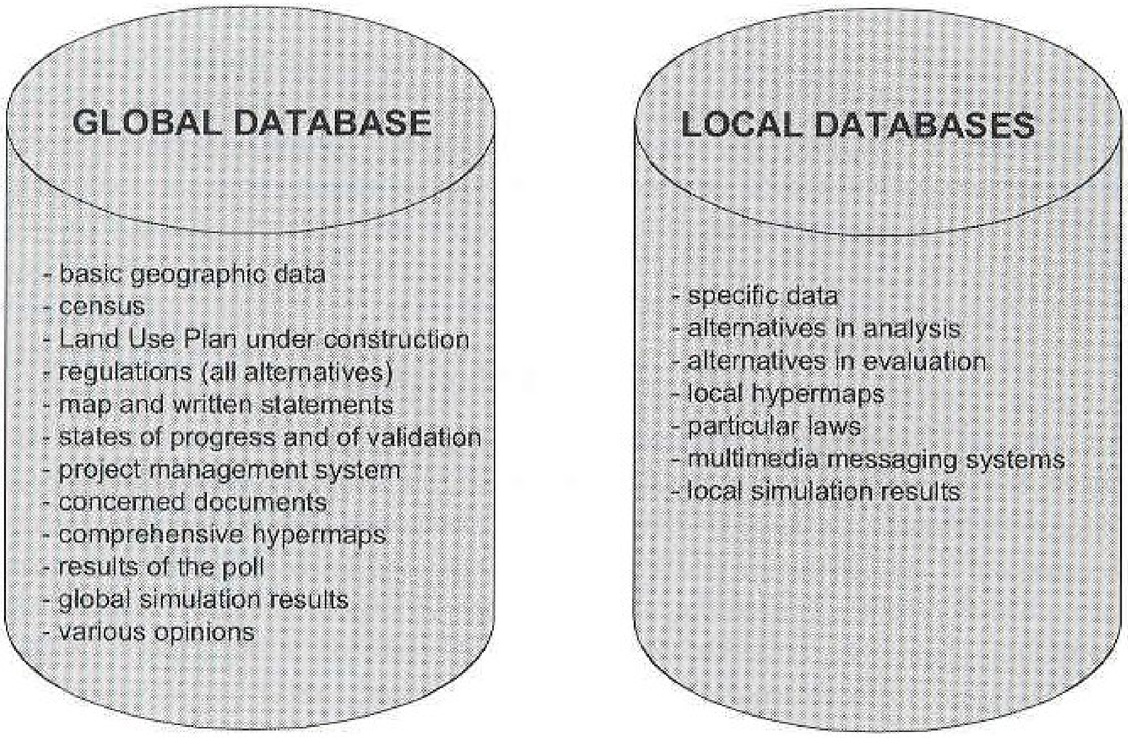

Figure 10.25 Contents of the central and the local databases for urban planning.

Contents of DBs for urban planning

- for cooperation : one central & multiple local DBs

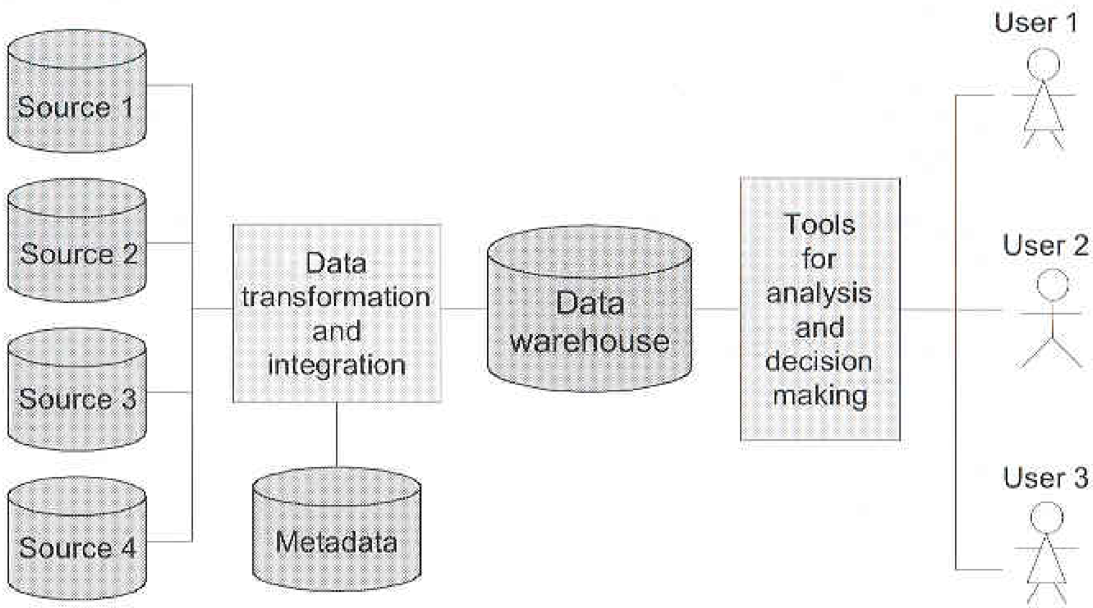

Figure 10.26 Structure of a datawarehousing system.
After Barquin and Edelstein (1997) with modifications.

10.6 Data warehousing
- a new possibility to organize & retrieve info

<u>Principles</u>

- to consider all data within an org & to make accessible to users
- taking the variety of data format into account, the data mining tools are different

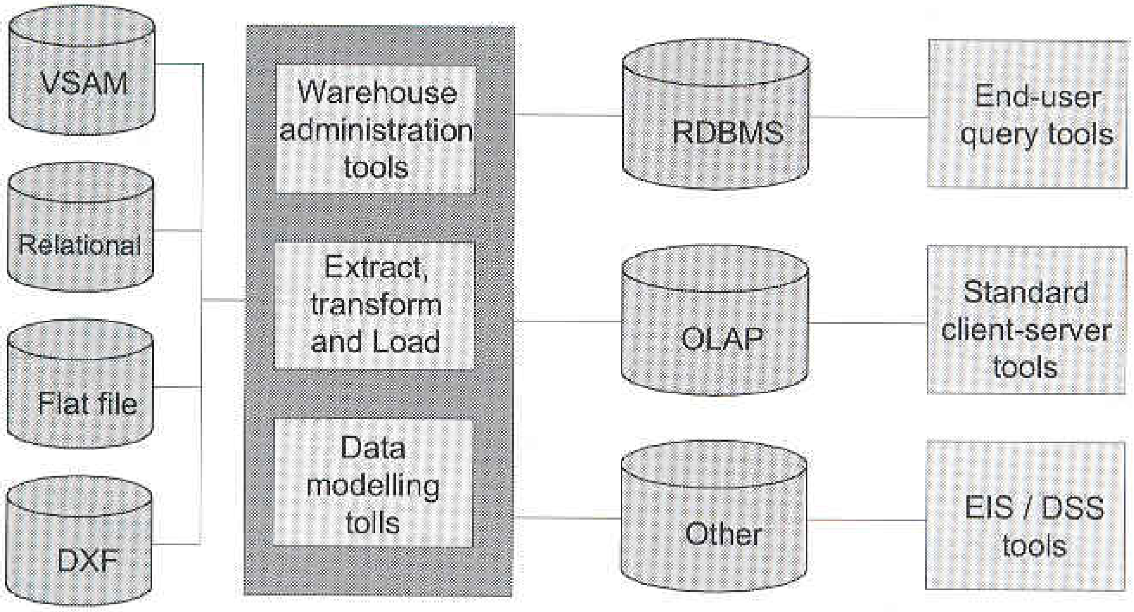

Figure 10.27 Components of a datawarehousing architecture system. After Barquin and Edelstein (1997) with modifications.

- components of a data warehousing architecture

  ∗ EIS : executive info sys DSS : decision support sys

  OLAP : on-line analysis procedure

Table 10.1 Comparing datawarehouses and operational databases

Datawarehouse	Operational database
Subject oriented	Application oriented
Integrated	Limited integration
Non-volatile	Continuously updated
Stabilised data value	Current data value only (generally)
Ad-hoc retrieval	Predictable retrieval

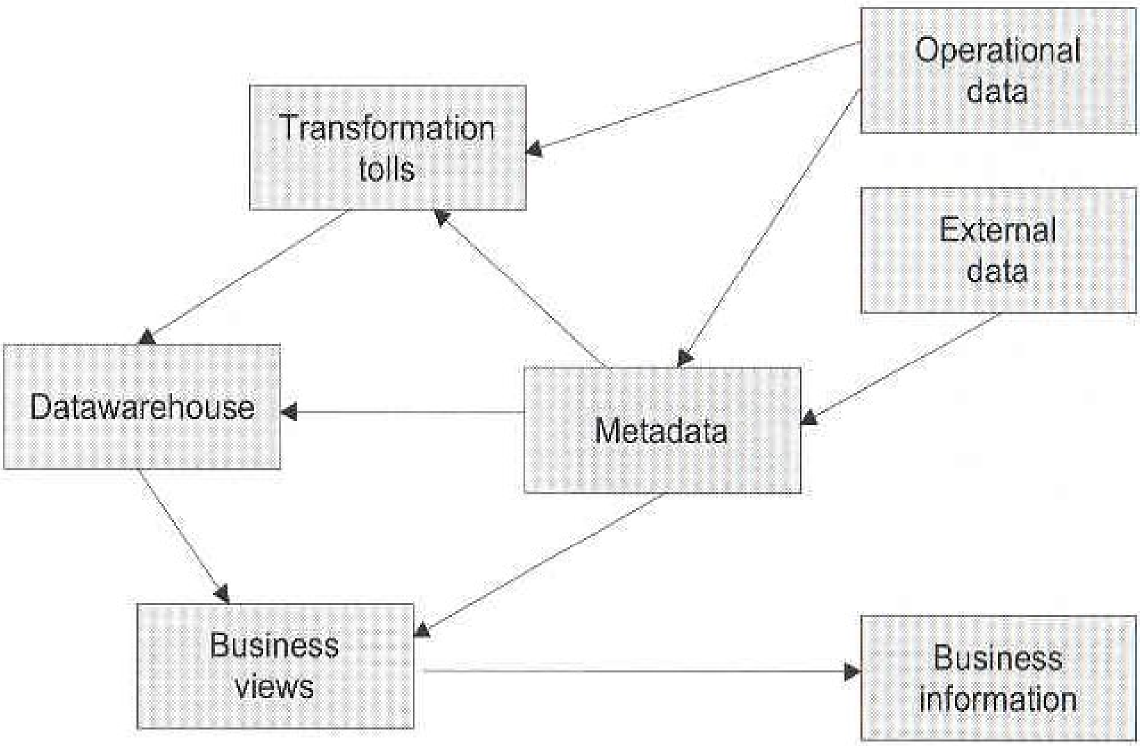

Figure 10.28 The process of transforming data into information by means of the datawarehouse technology.
According to Bontempo and Zagelow (1998) with modification.

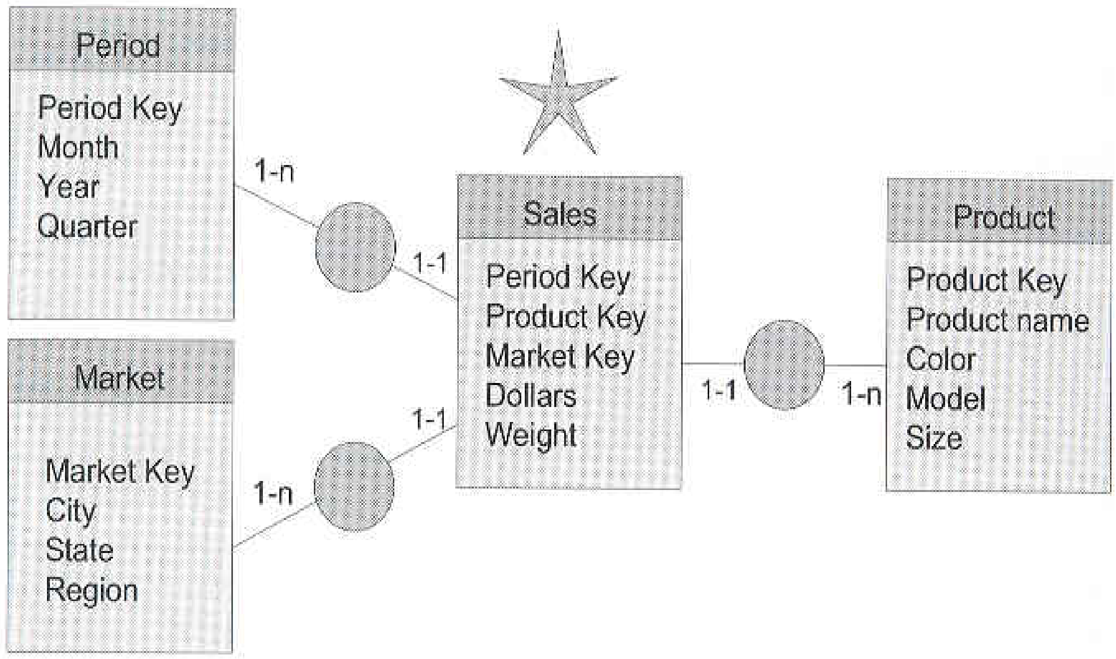

Figure 10.29 Simple star schema: one fact table (Sales), three dimension tables (Product, Period and Market).
According to Livingston and Rumsby (1997).

Star structure & data cubes

- star schema articulates a design strategy that enforces clear & simple relationships between all info in data warehouse

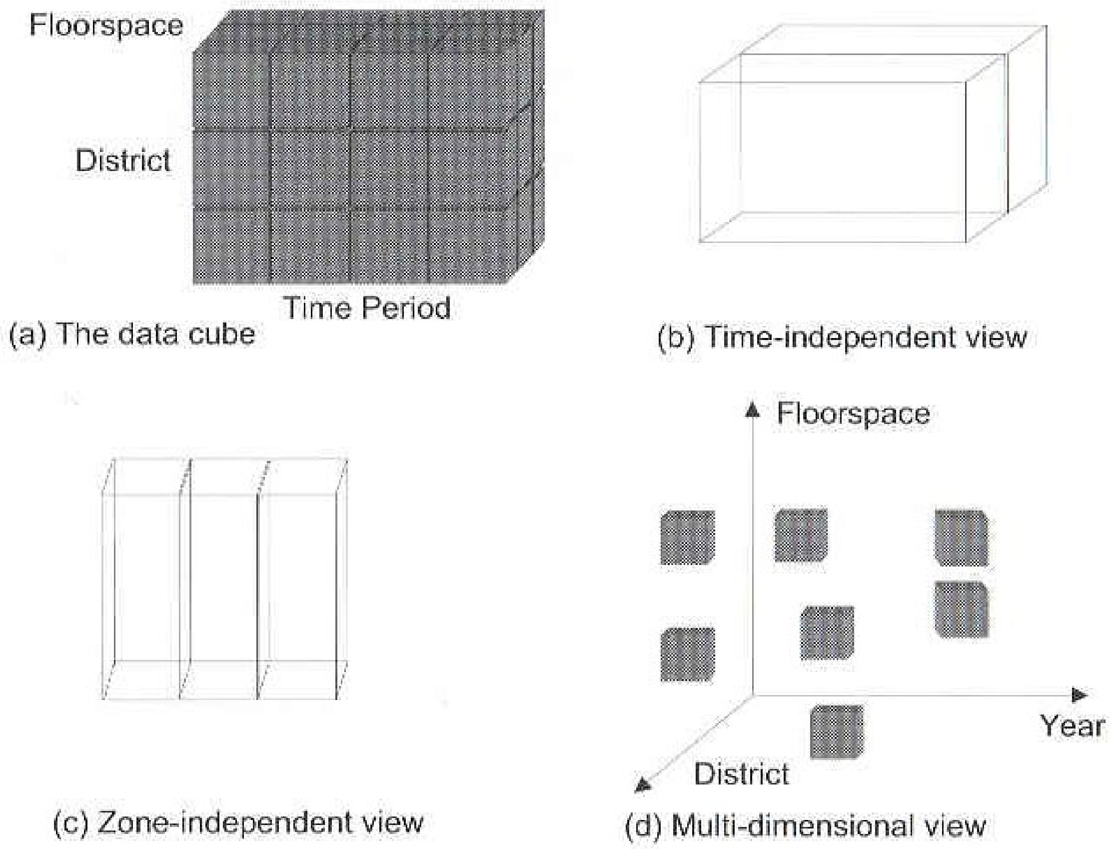

Figure 10.30 Example of a datacube for accessing a datawarehouse for floorspace analysis. (a) The datacube. (b) A time-independent view. (c) A zone-independent view. (d) The multi-dimensional view.

- data cube for accessing a data warehouse

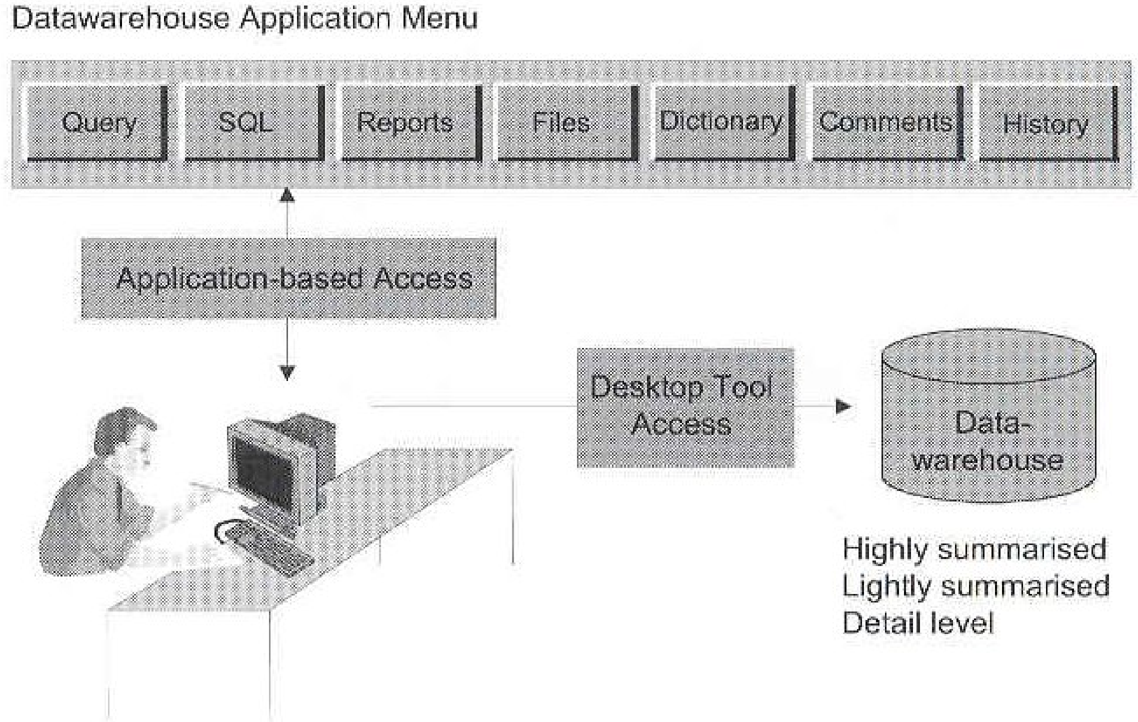

Figure 10.31 Example of a datawarehouse implementation. From Barquin and Edelstein (1997), with modification.

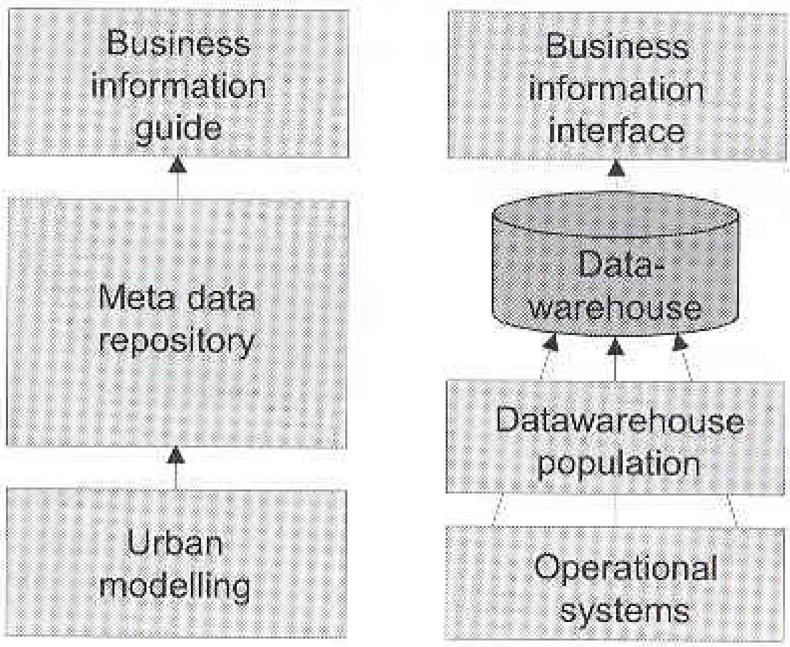

Figure 10.32 Architecture of a datawarehouse system. According to Devlin (1997) with modifications.

<u>Data warehouse architecture</u>

- several different architectures

  centralized data warehouse : simple architecthre

  data warehouse & data marts : data marts contain narrower scope of data

  distributed data warehouse : connected via network

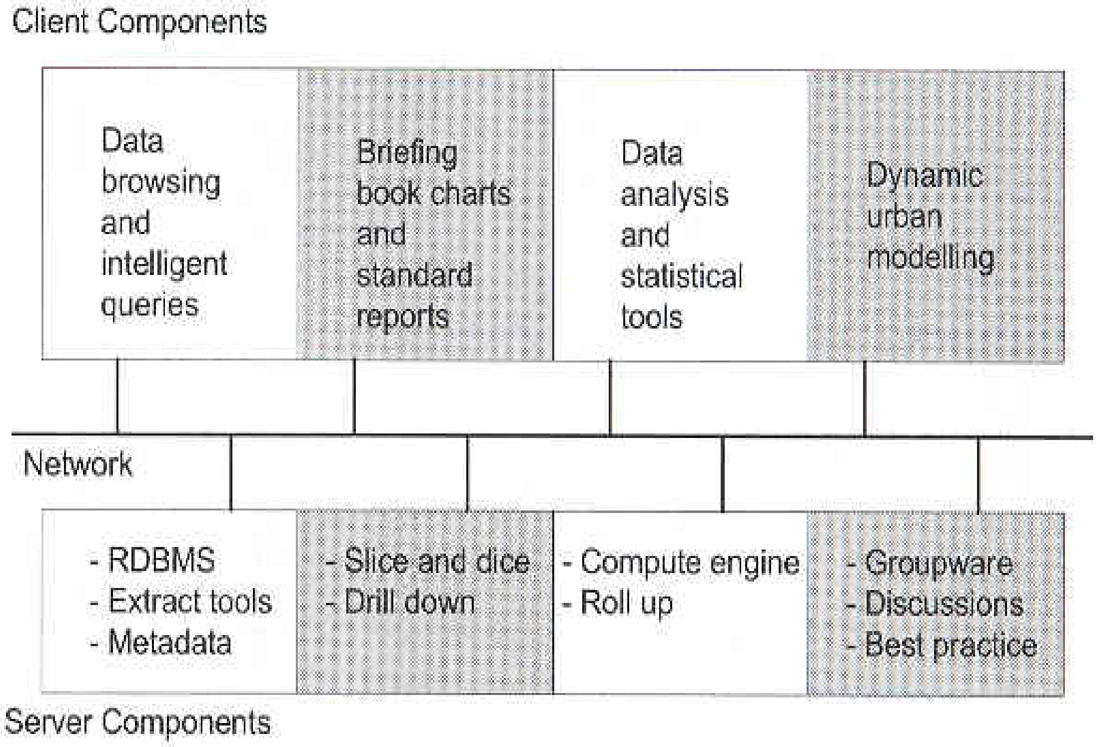

Figure 10.33 Potential datawarehousing components. According to Neal (1997) with modifications.

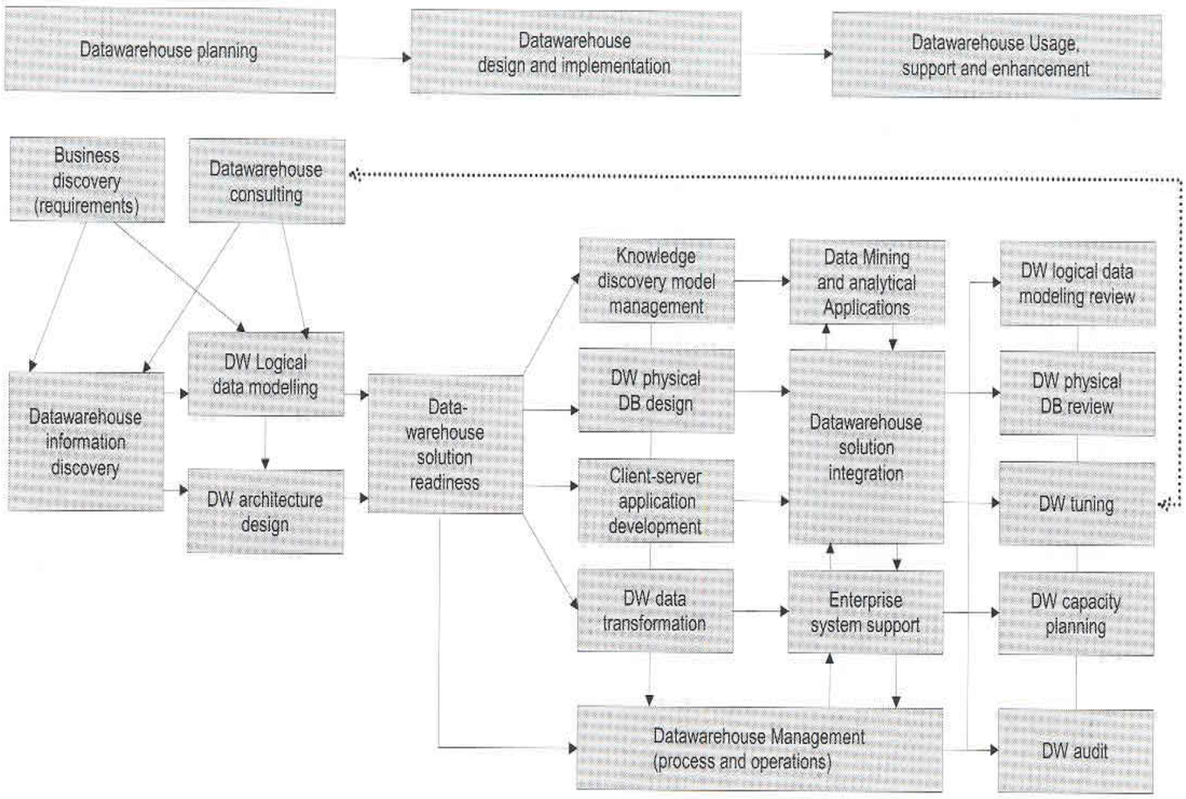

Figure 10.34 Building and managing a datawarehouse.
According to Gardner CACM Sept 1998 p. 57 with permission.

Methodology

- methodology to design a data warehouse
  a. data warehouse planning
  b. data warehouse design & implementation
  c. data warehouse usage, support & enhancement

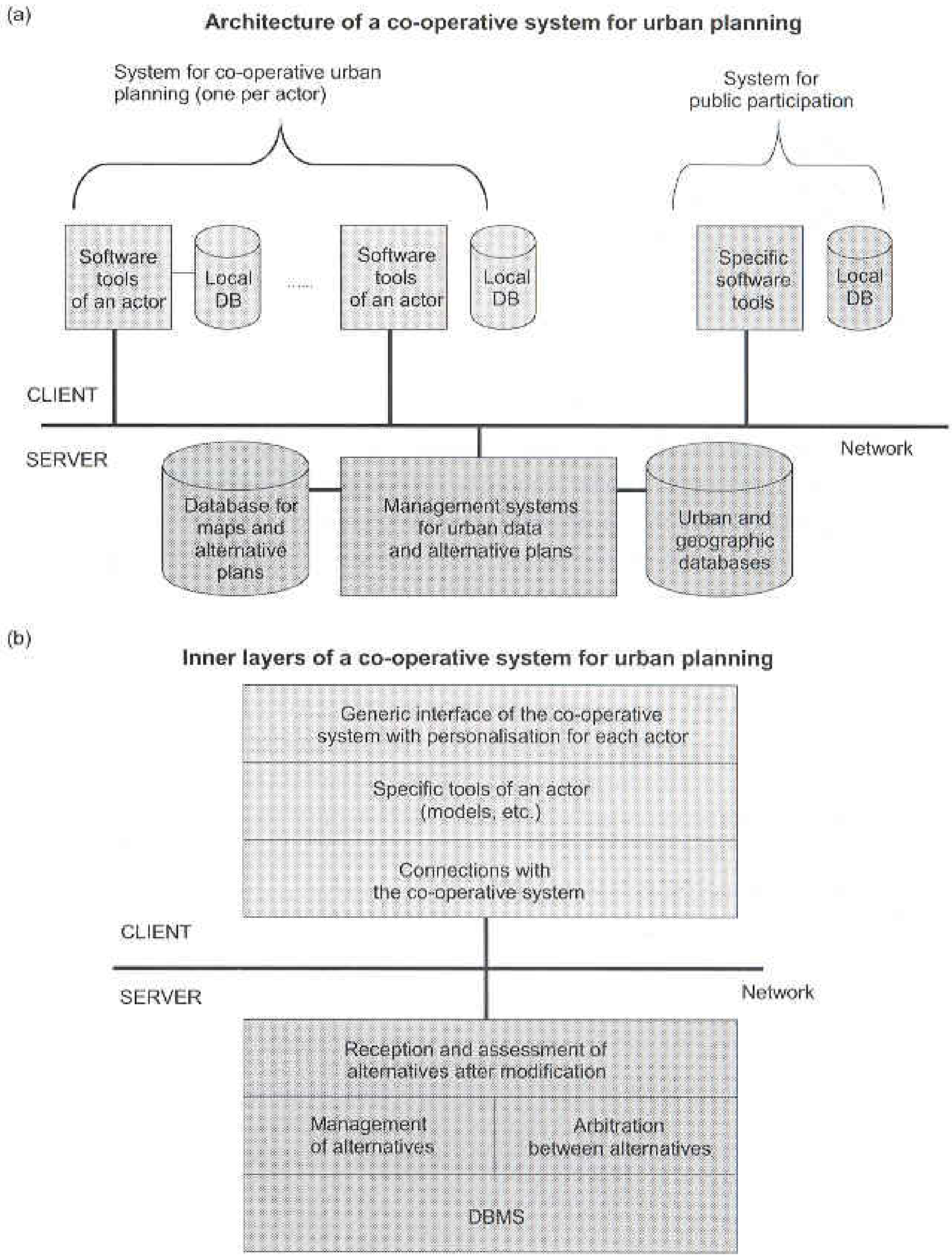

Figure 10.35 Architecture of a co-operative system for urban planning. (a) Principles. (b) Inner layers.

<u>Data warehousing & urban planning</u>