Chapter 2 Design Methodologies for Info Systems

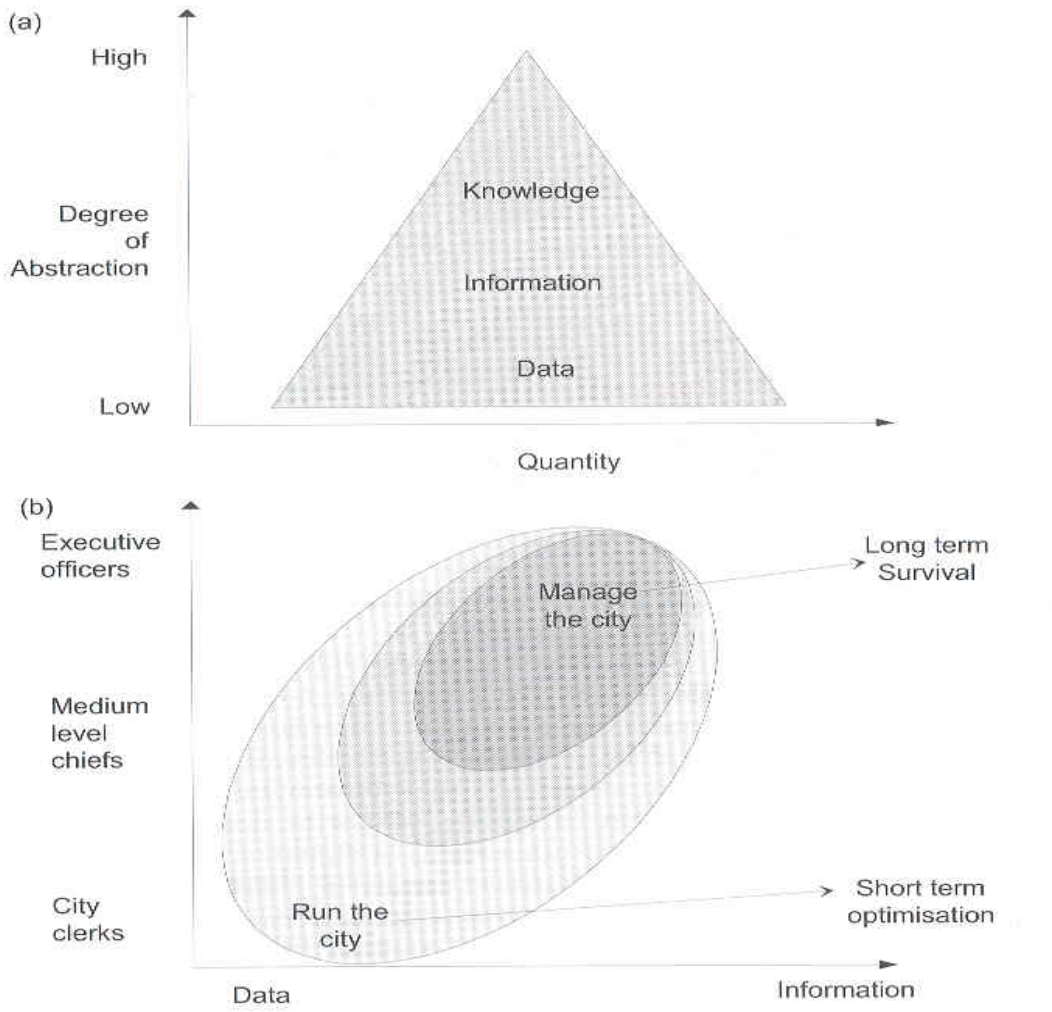

Figure 2.1 From data to information and knowledge. (a) Various levels of abstractions (from Turban and Aronson, 1998 with modifications). (b) From running to managing a city.
From Devlin, 1997 with modifications.

2.1 Data, info, knowledge & meta data

- Data : numbers, strings, Booleans, sounds, images ..
- Info : organized data, has a meaning
- Knowledge : results of analyzing & synthesizing info, derived info
- Meta data / meta info : data about data/ info about info

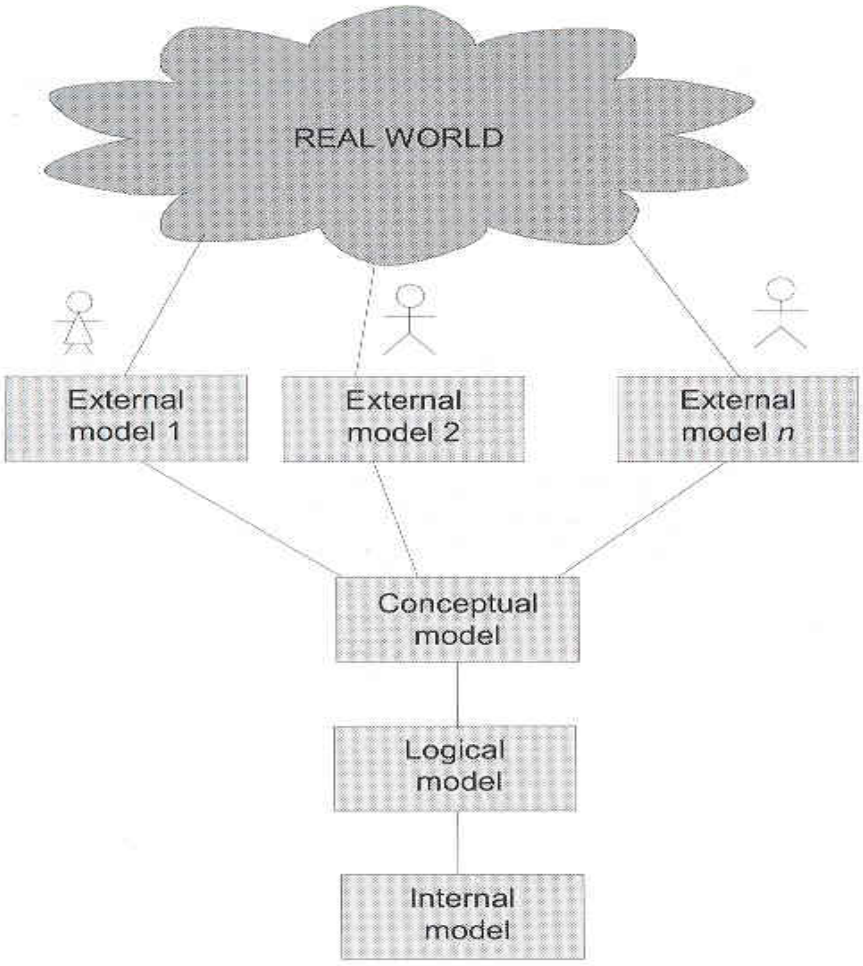

Figure 2.2 Data modelling levels (ANSI/X3/SPARC 1978).

2.2 Modeling levels (external, conceptual, logical, internal)

- External : users define their own subset of the real world
- Conceptual : schematic representation of phenomena & their relationship

  deals w/ only info content of the database
  usually use entity-relationship approach
- Logical : data structure, relational/object oriented/object-relational
- Internal : represent the way info items are stored at bit/byte level

  deals w/ storage devices, file structures, access methods ..

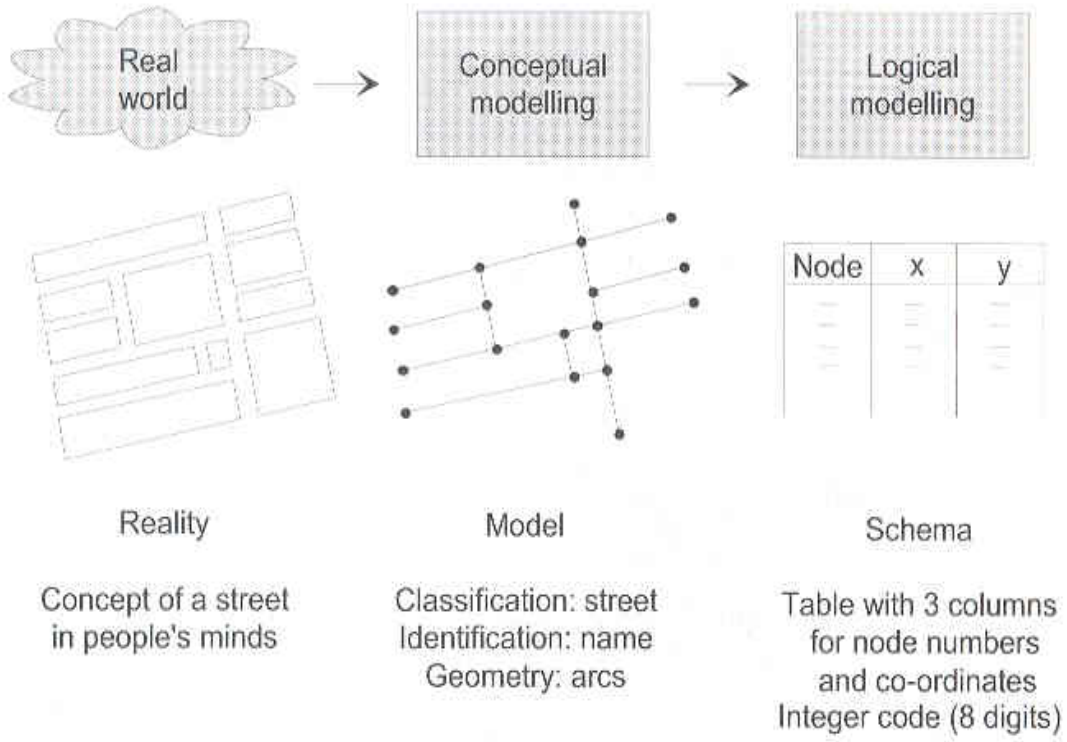

Figure 2.3 Example of steps for modelling streets. Bartelme 1996, with modification.

Example of steps for modeling streets

external model -> conceptual model -> logical model

(real world) (classification, identification, geometry) (data org schema ex. relational DB)

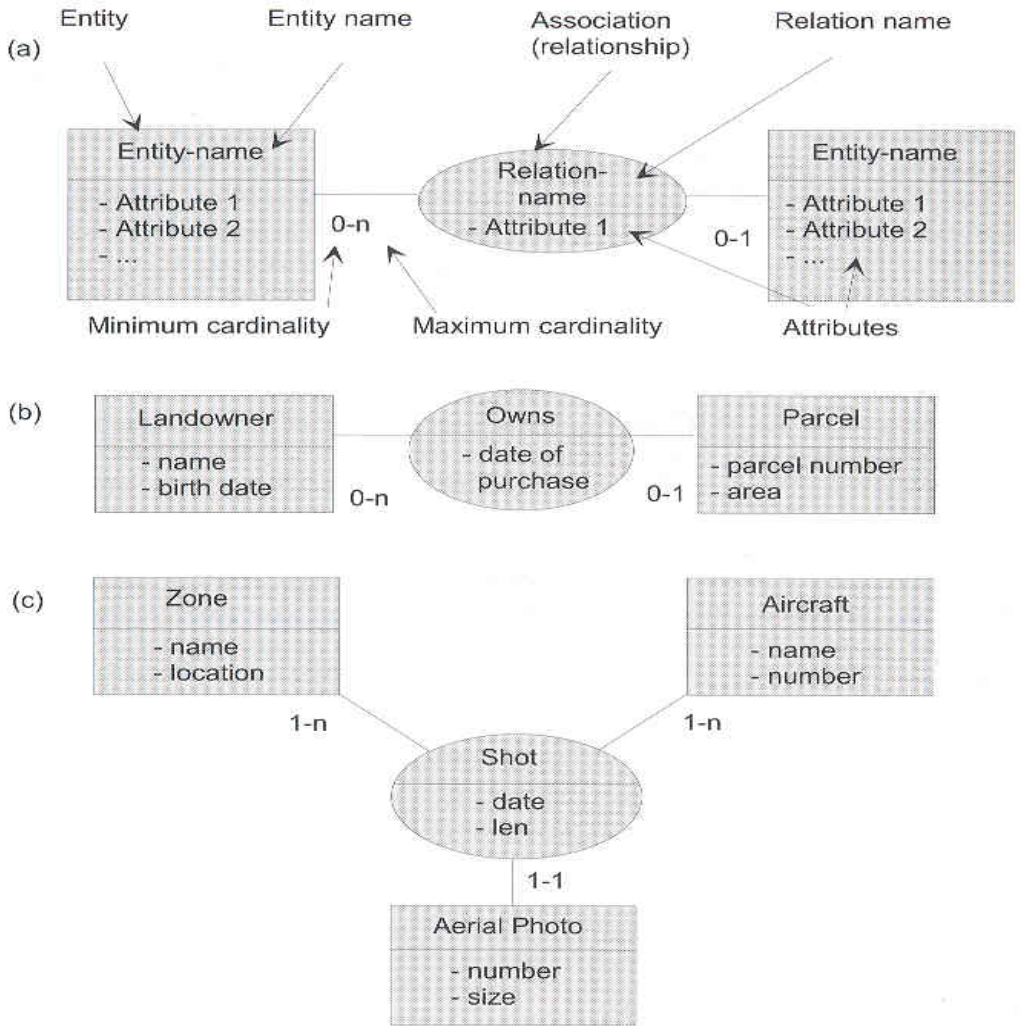

Figure 2.4 Examples of entity-relationship diagrams. (a) Nomenclature for entity-relationship diagrams. (b) Example of a binary relationship. (c) Example of a ternary relationship.
From Laurini and Thompson (1992).

2.3 Conceptual modeling : entity-relationship approach

- E-R model : a means to organize & schematize info

  basic component : entities (classes of entities), relationships, attributes, cardinalities, integrity constraints

  * relationships can be multiple (binary, ternary ..)

  * integrity constraint is a predicate that must be matched

    ex) attribute values, attribute definition, cardinality ..

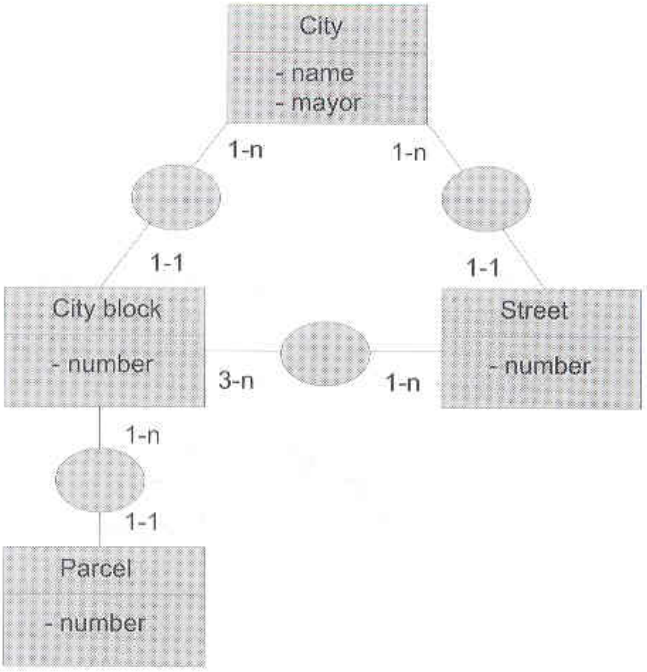

Figure 2.6 The entity-relationship diagram for land parcels and city blocks: a city can have several city-blocks and streets. Each block is surrounded by a minimum of three streets and can have a minimum of one parcel.

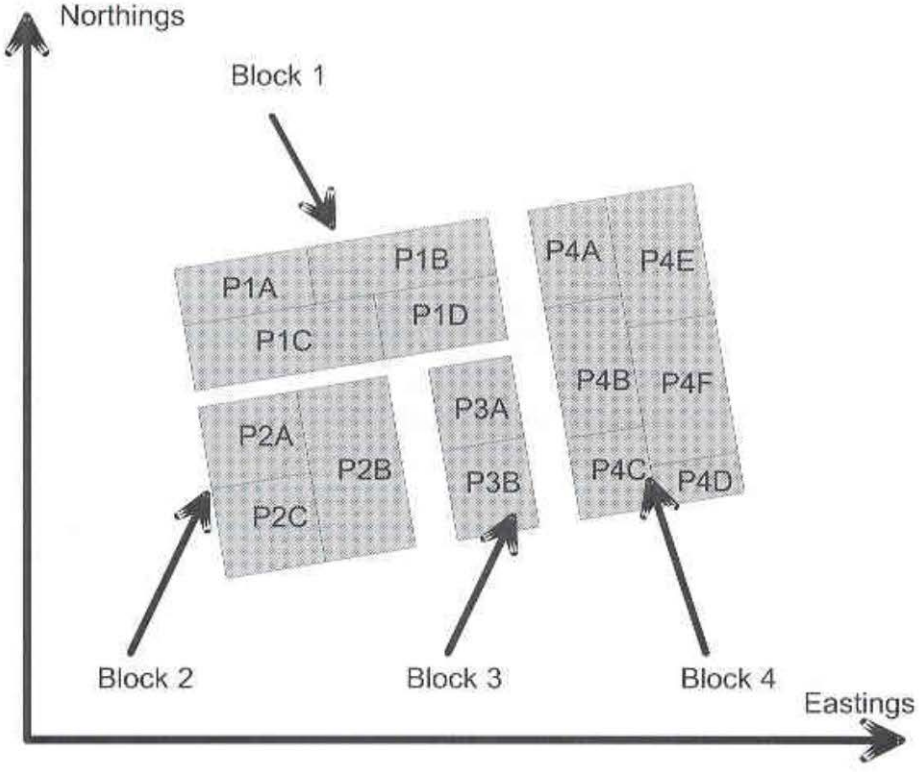

Figure 2.5 An example of city blocks and parcels.

Table 2.1 Example of a table for land use planning

R	*Zone-number*	*Area*	*Landuse*
	34	4578	commercial
	35	3471	residential
	36	9065	industrial

2.4 Logical modeling: relational databases

- Relational DB : collection of relations via tables

conventional form of expression

R (Zone-number, Area, Landuse) : single relation for landuse planning

-> its table containing an instance of this example(Table 2.1)

columns: fields (attributes) – data type, format, unit need decided

identifier attribute : key

rows: records/tuples (particular instances of an entity)

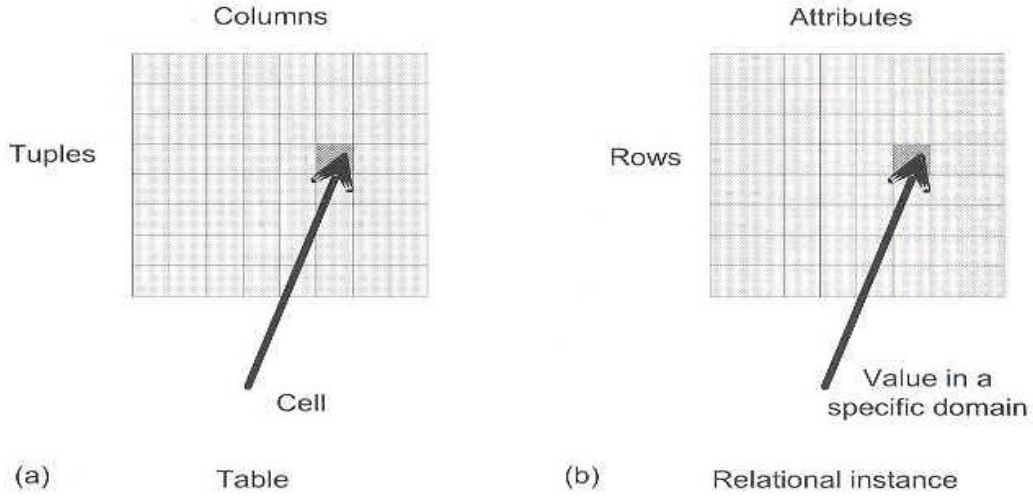

Figure 2.7 Nomenclature for relational tables. (a) Structure of a table. (b) Relational instance with a particular value.

Table 2.2 Example of a more complex relation

PURCHASE	*Plot_ID*	*Owner_ID*	*Purchasing_date*
	5678	45	13-03-76
	4567	67	16-05-81
	1208	85	01-07-78
	3089	60	20-12-80
	3218	49	31-01-82
	.....	.....	.....

- Example of relation schema

PURCHASE (Plot_ID, Owner_ID, Purchasing_date) (-> Table2.2)

PURCHASE : relation, Owner_ID,Purchasing_date : attributes

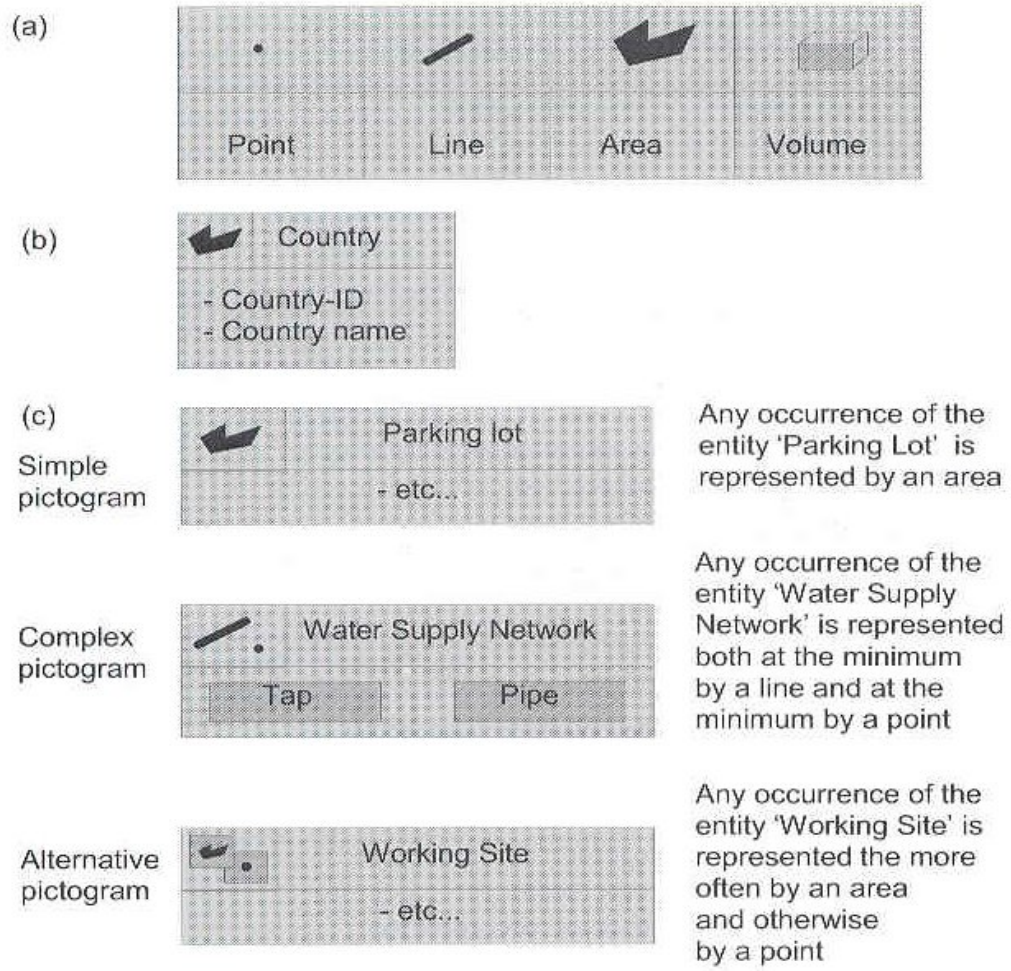

Figure 2.8 Spatial pictograms. (a) The four basic spatial pictograms. (b) An entity with its spatial pictogram. (c) Examples of simple, complex and alternative spatial pictograms.
From Caron *et al.* (1993) with modifications.

2.5 Spatial pictograms

- Spatial pictogram : need to represent geo-entity in the E-R approach

- Four types : Point, Line, Area, Volume

- Examples

  single case, complex case, alternative case

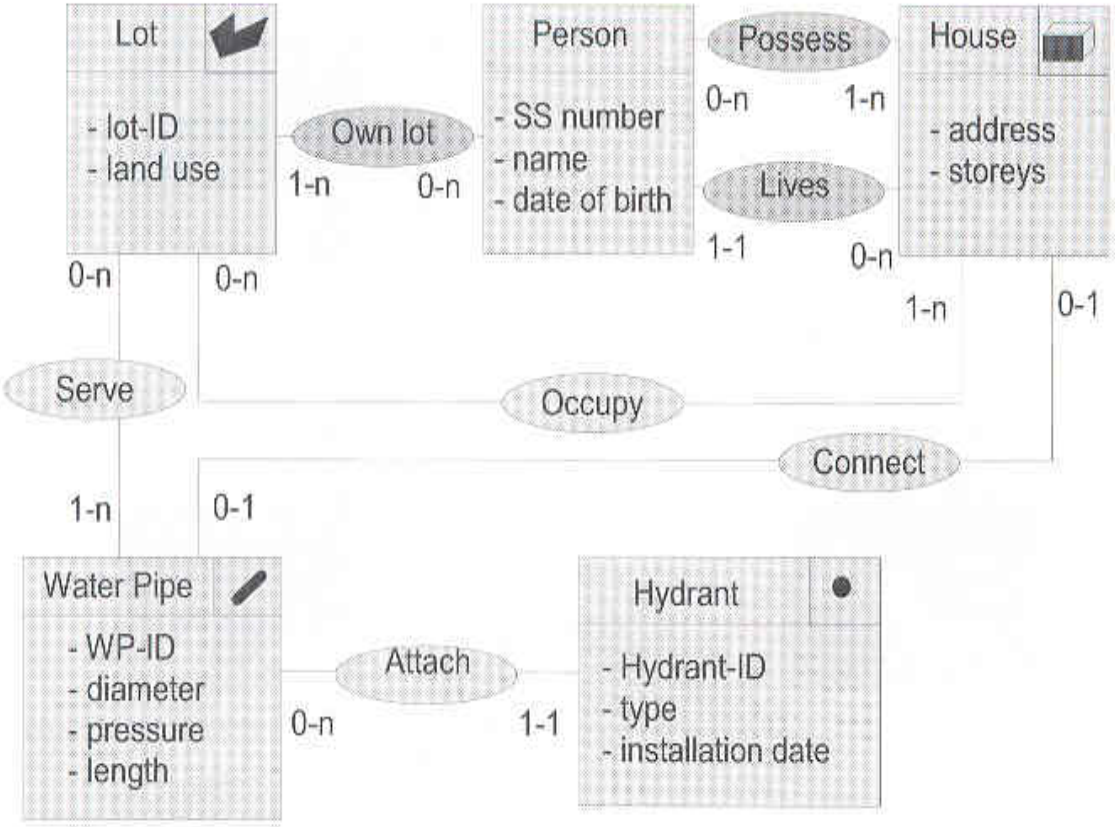

Figure 2.9 Example of an entity-relationship diagram with pictograms. From Caron *et al.* (1993).

- Example of an E-R diagram w/ spatial pictograms

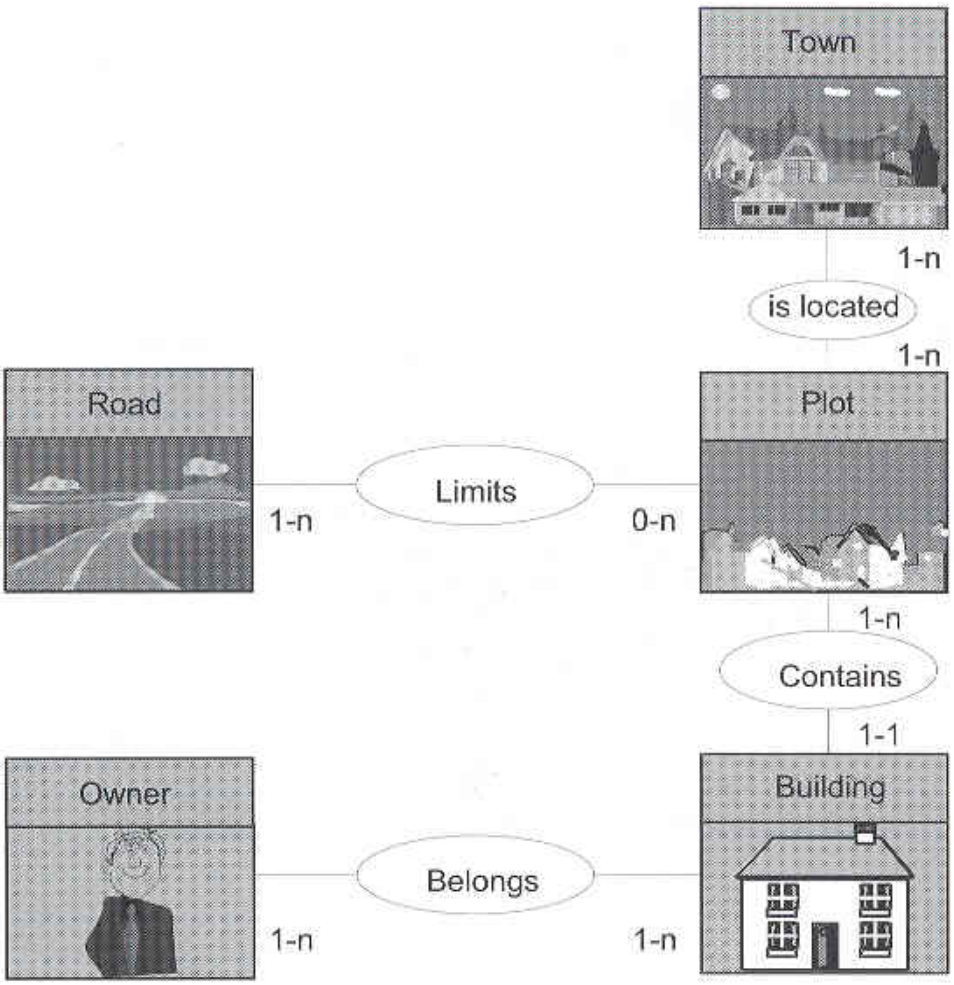

Figure 2.10 Example of an entity-relationship model with entity pictograms.

2.6 E-R relationship w/ entity pictograms

- Entity pictogram : to increase readability & abstraction

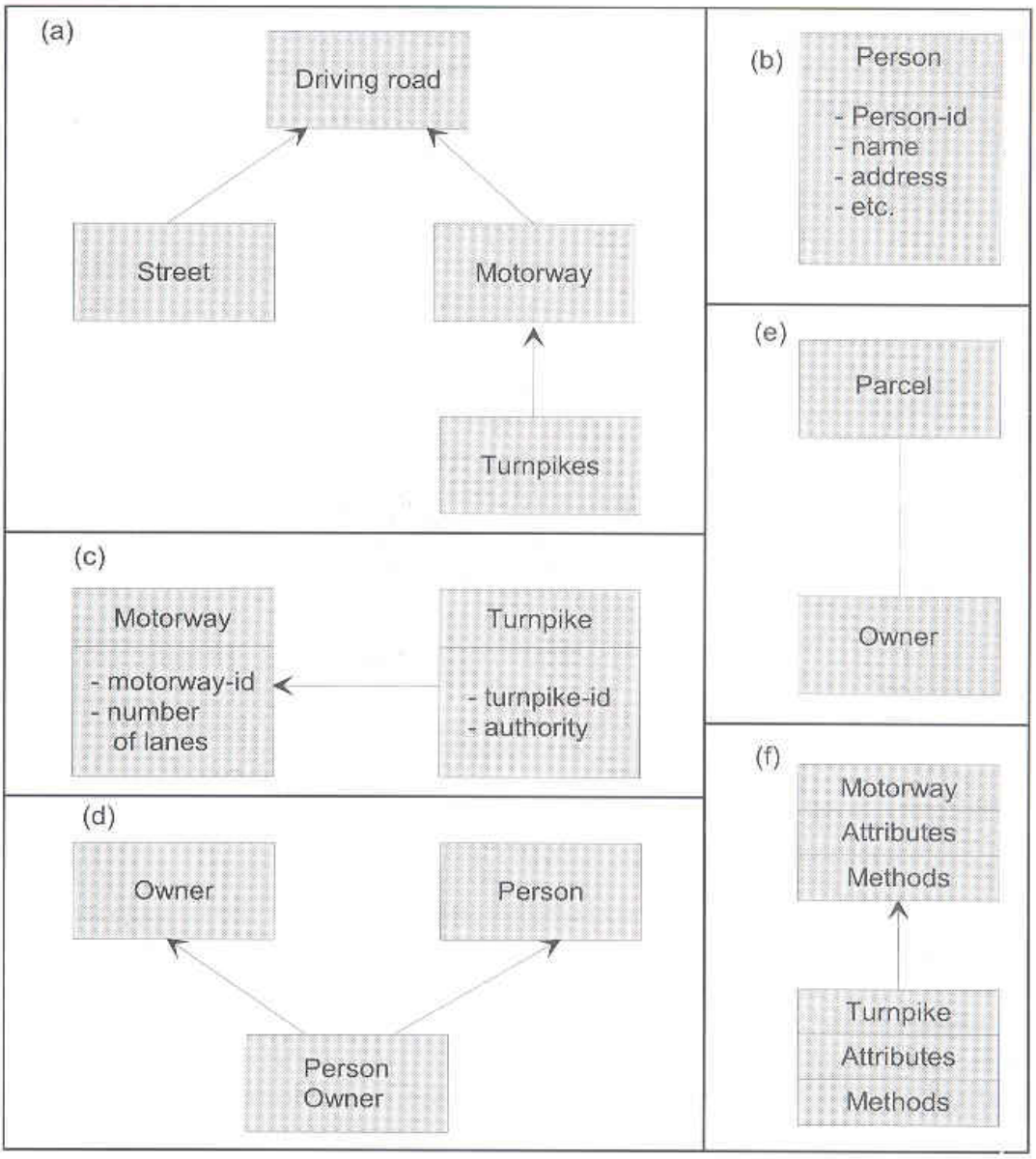

Figure 2.11 Example of object-oriented concepts. (a) Classes and subclasses. (b) Attributes. (c) Attributes in inheritance hierarchy. (d) Multiple inheritance. (e) Links between classes. (f) Inheritance of methods.

2.7 Object-oriented spatial DBs

- OODBs : to improve modeling of the real world

  class : a collection of objects w/ the same behavior

  instance : a particular occurrence of an object for a given class

- Class can define subclasses within it (Fig 2.11 a)

  class -> subclass / class -> superclass

  ex. CLASS Motorway : SUBCLASS_OF Driving road

  CLASS Street : SUBCLASS_OF Driving road

  CLASS Turnpike : SUBCLASS_OF Motorway

- Class has attributes

  ex. CLASS Turnpike : SUBCLASS_OF Motorway

  { Turnpike-name : string
  Toll-free : money }
  * money can be a class

- Object in a subclass inherits all attributes of the superclasses
- Objects may be linked in different ways in a DB
- Object includes methods (operation on the data)

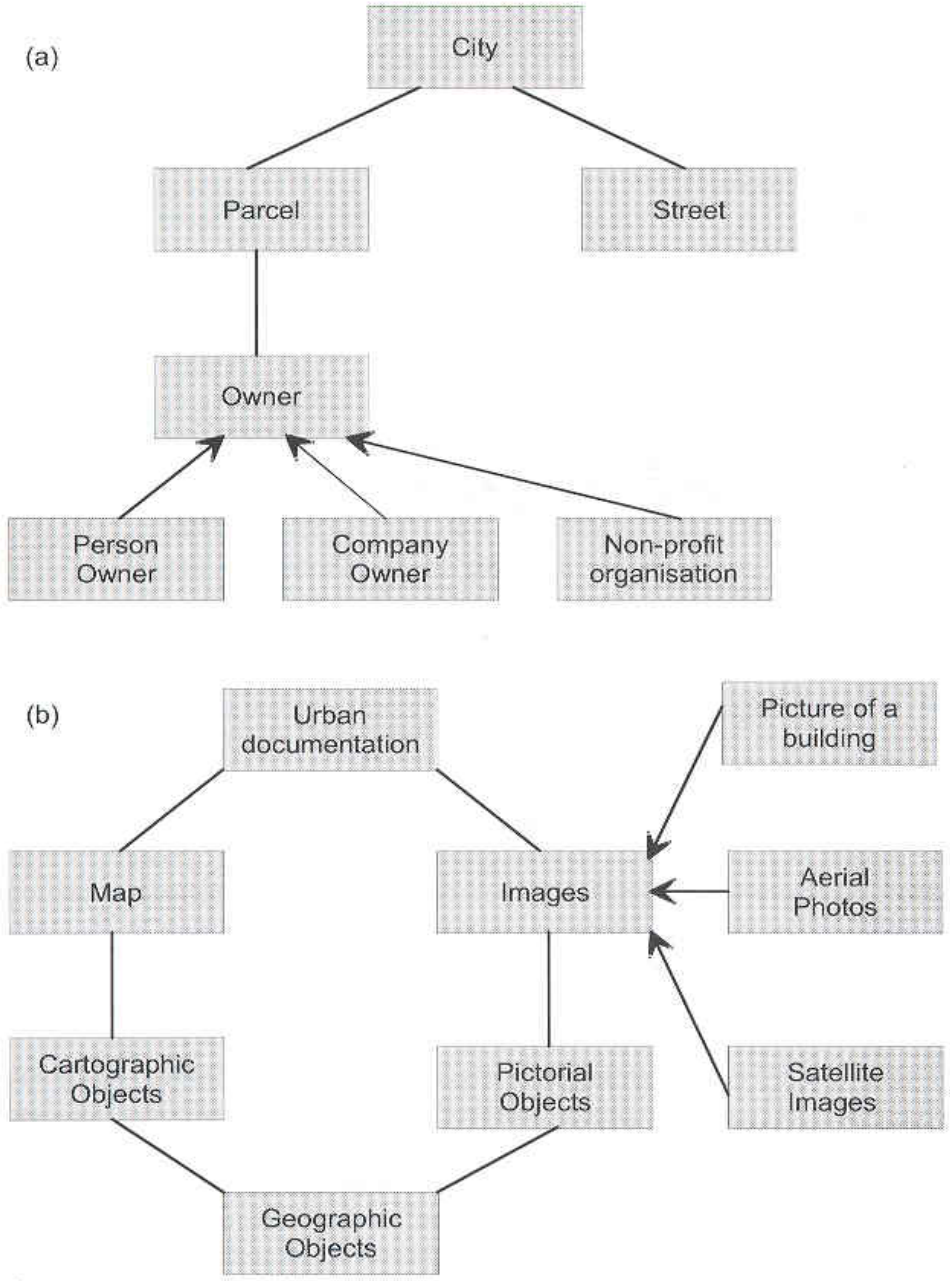

Figure 2.12 Examples of object-oriented modelling. (a) In urban planning. (b) For urban documentation.

- Examples of object-oriented modeling

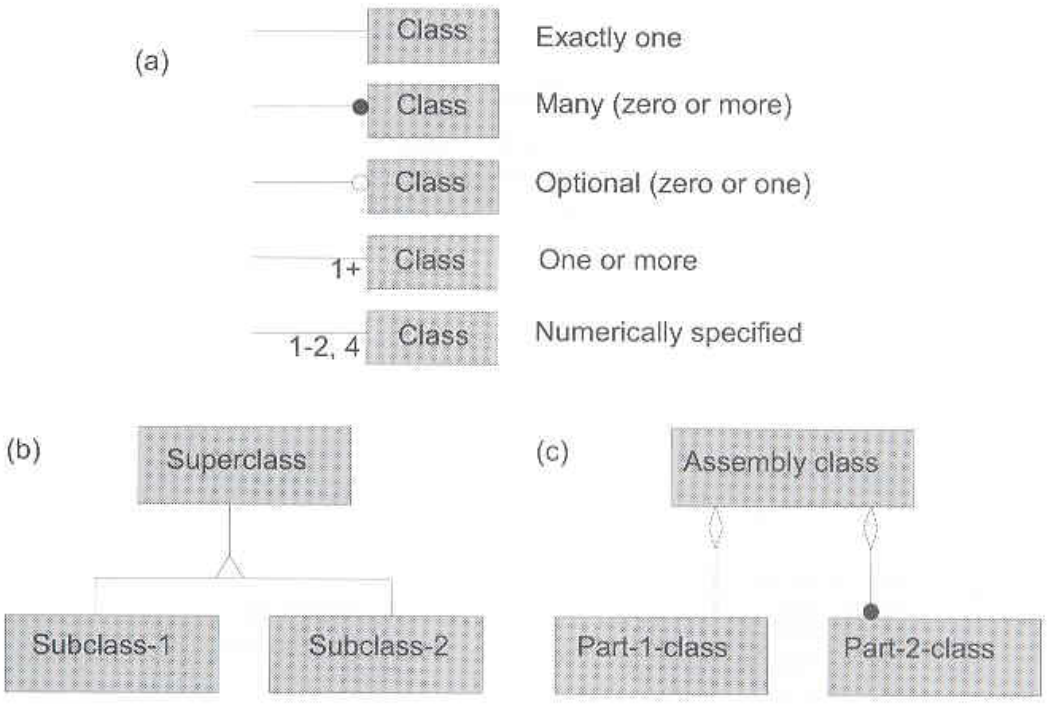

Figure 2.13 The OMT/UML formalism. (a) Various types of associations depending on cardinalities. (b) The specialisation in subclasses. (c) The aggregation formalism emphasising that Part-1-class has a unique instance whereas Part-2-class has several (Rumbaugh *et al.* 1991).

2.8 Object-oriented DB design methodology

- Object-oriented DB design is possible by OMT / UML

  * OMT : object modeling tech UML : unified modeling tech

- OMT / UML allows one :

  to define classes w/ attributes & methods

  to define inheritance

  to define aggregation & specialization

  to define association between classes, and so on.

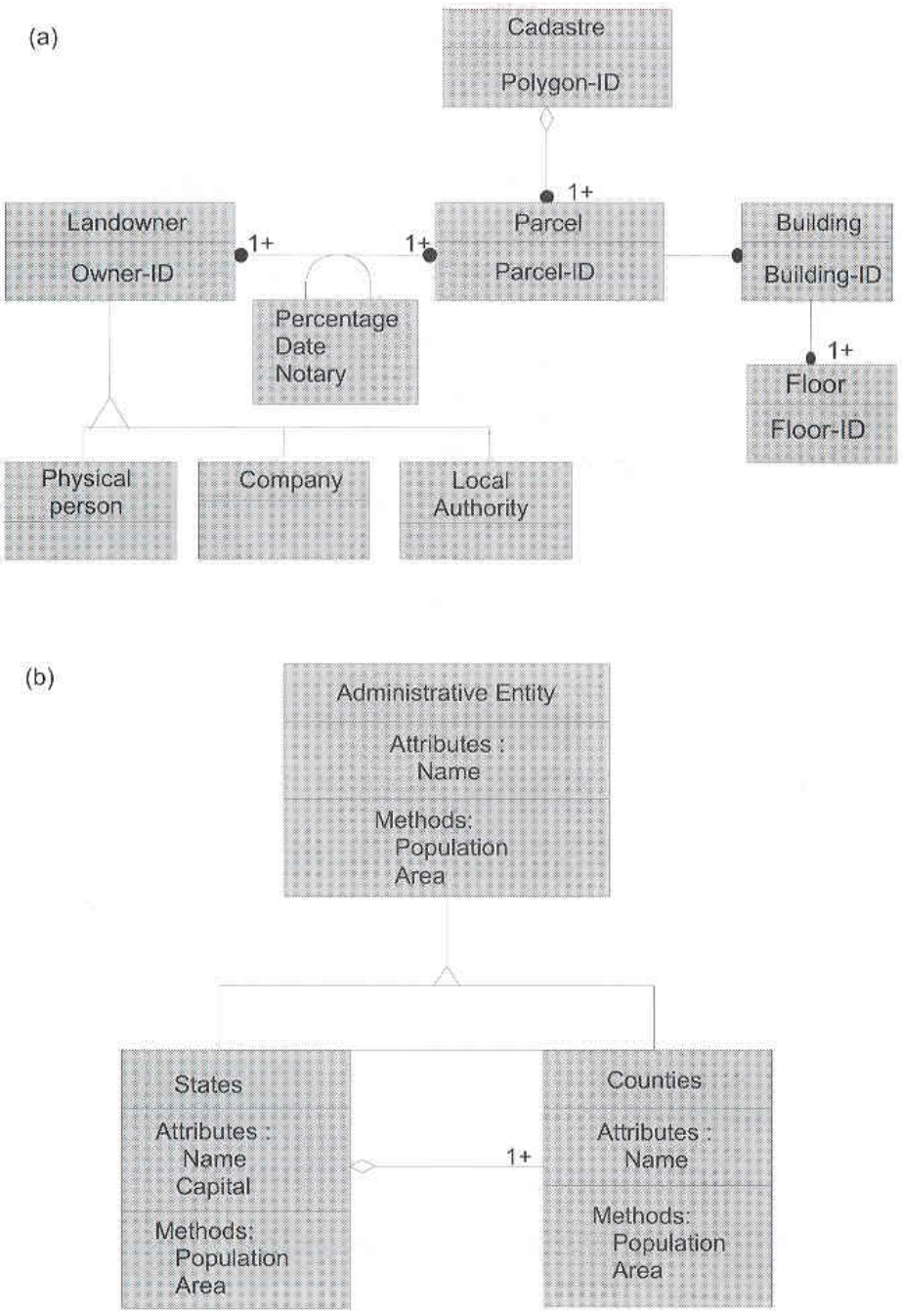

Figure 2.14 Example with OMT/UML formalism. (a) Ownership and cadastre without methods. (b) Administrative bodies in the US with their relations with methods.

- Example of OMT / UML formalism

  * semi-circle represents attributes of a relation

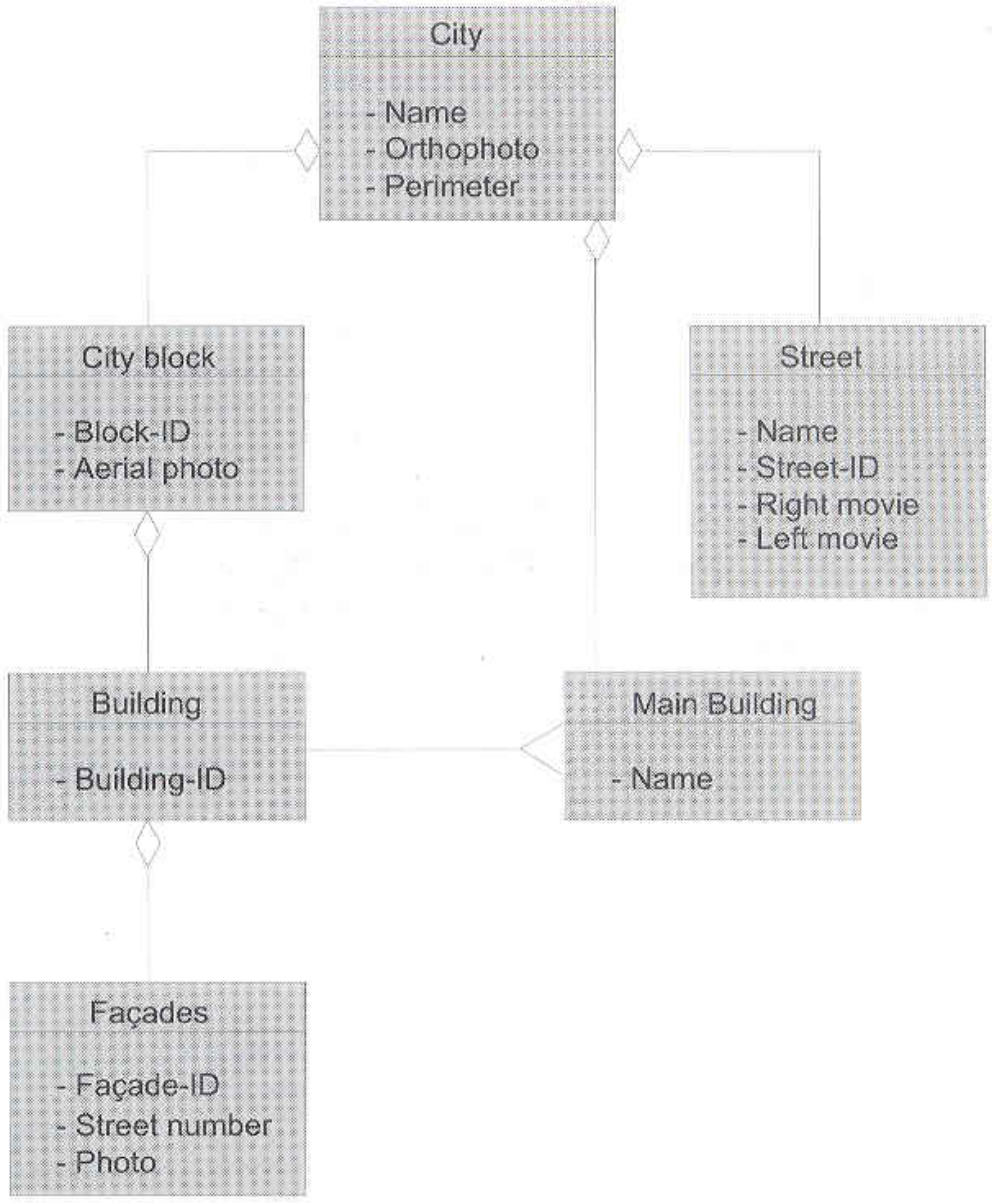

Figure 2.15 An urban database structure for storing façades of all buildings.

- OMT / UML model for a DB storing facades

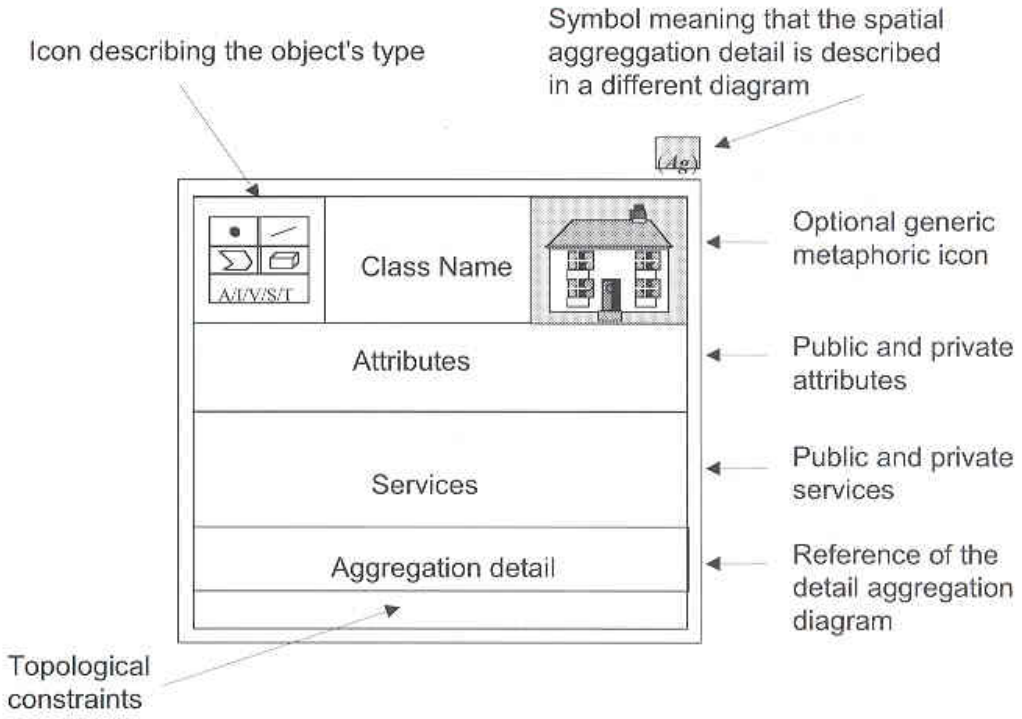

Figure 2.16 Graphic description of spatial meta-class.

2.9 Presentation of OMEGA

- OMEGA : based on OOA, MODUL-R, CONGOO, OMT-Booch

  -> leads to an operational description of GUI, org & architecture

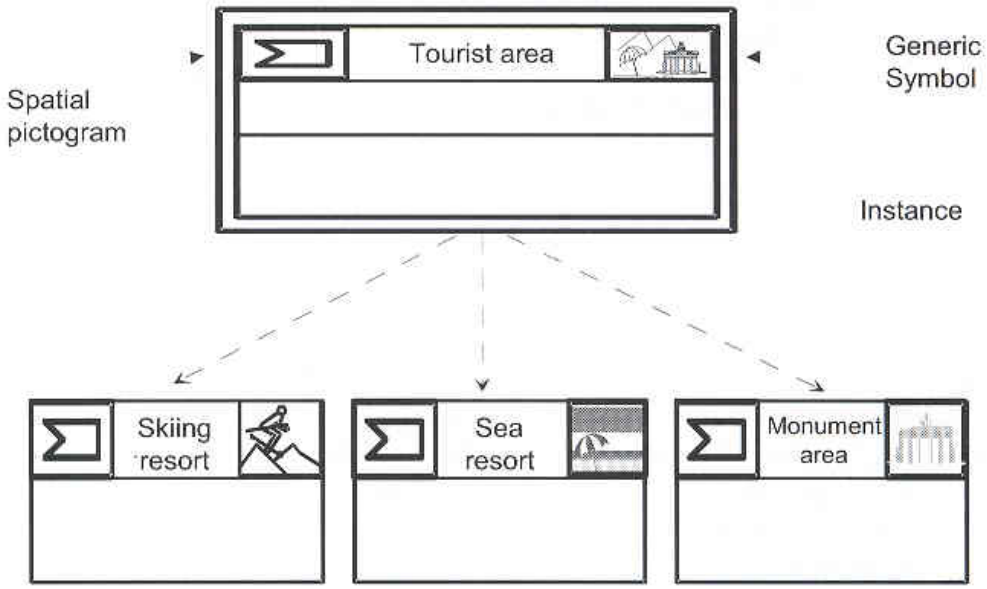

Figure 2.17 An example of a representation of meta-classes and their instances in OMEGA formalism (Lbath 1997, Lbath *et al.* 1997).

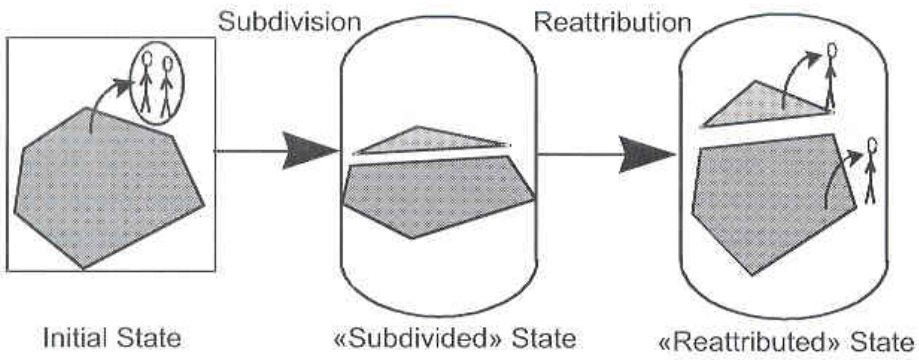

Figure 2.18 Example of dynamic representation of object in function of its current state.

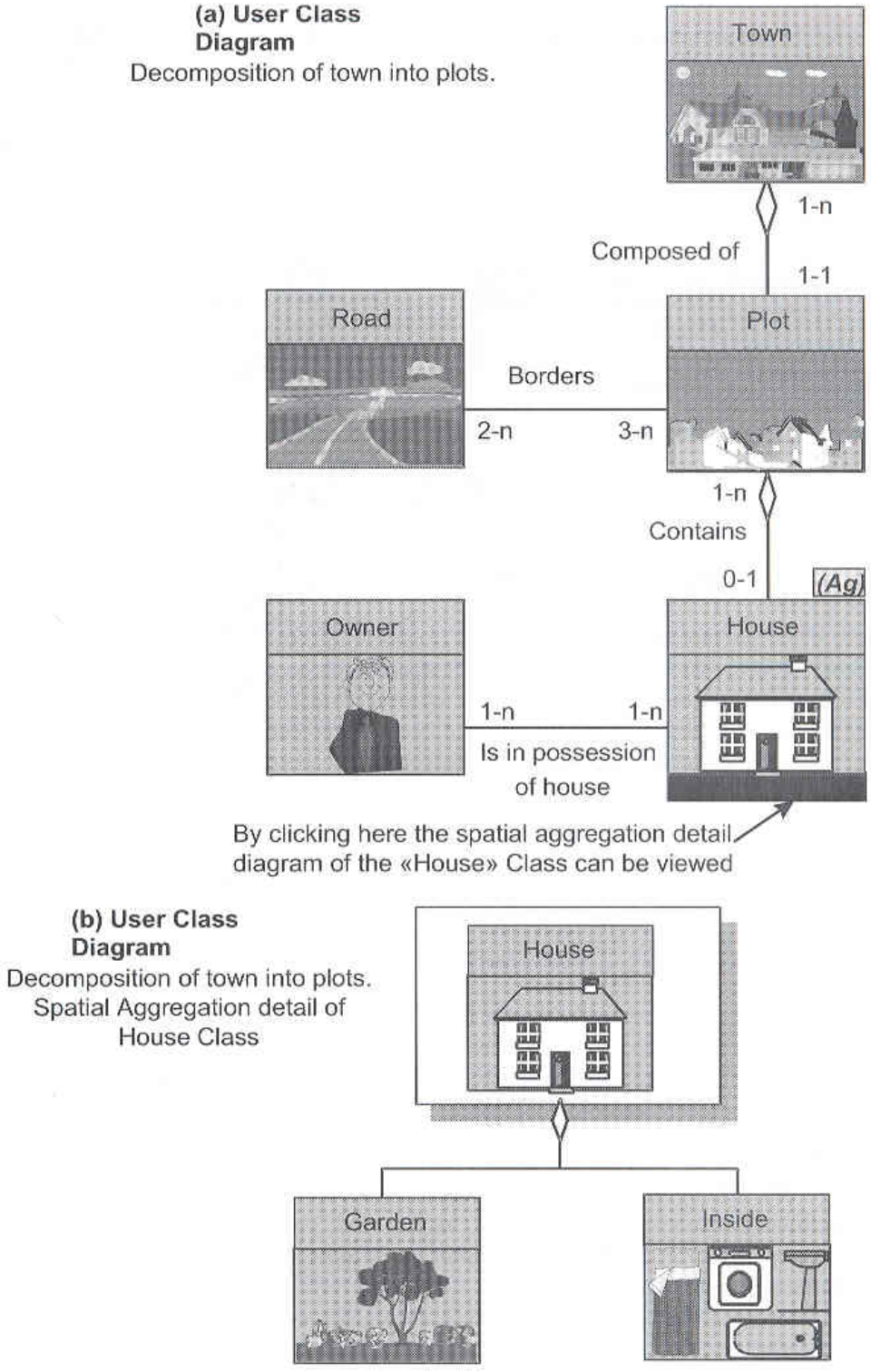

Figure 2.19 An example of a user class diagram with OMEGA formalism.

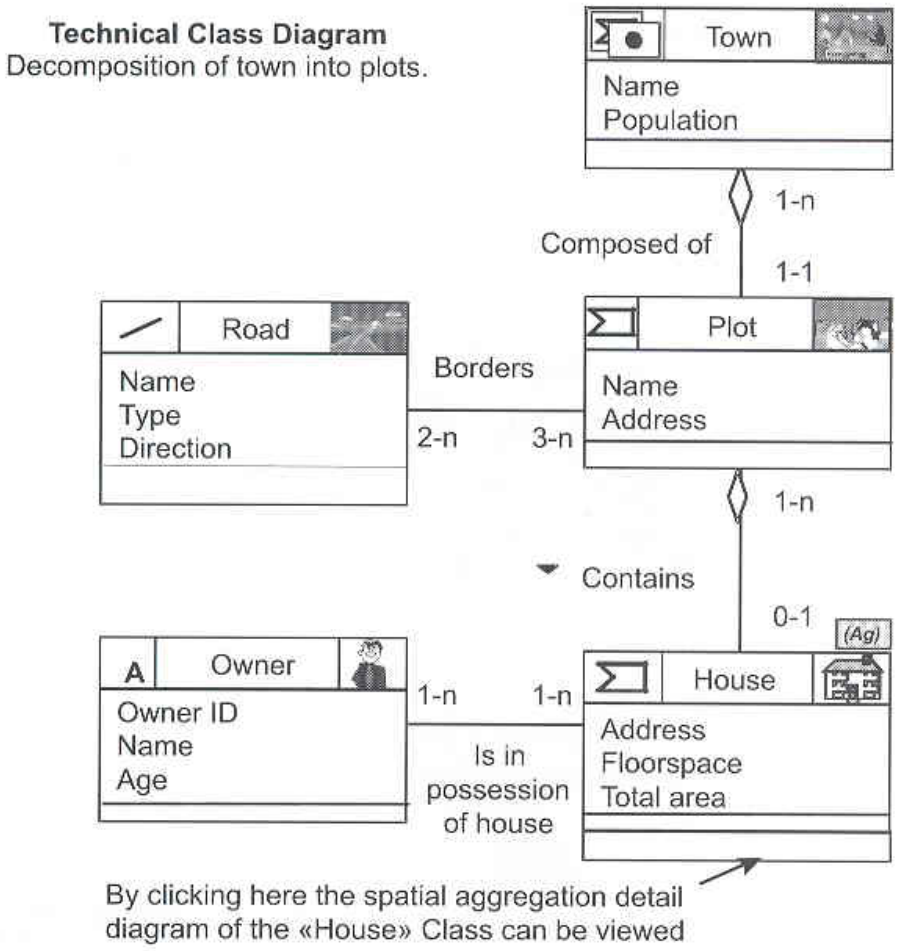

Figure 2.20 An example of a technical class diagram with OMEGA formalism. Metaphoric icons are replaced by a technical representation of objects.

2.10 Extended relational formalism

- Extend the relational model to integrate some object-oriented characteristics

  -> introduce subtables in relational tables

  : *R1 (a, b, c*)* *c** attributes are a list of values

  ex. *OWNER (Owner_ID, Name, Address, Property*)*

  a subtable can be nested in another subtable

  ex. *PERSON (Person_ID, Last_name, (kind_names, toys*)*)*

  geoprocessing ex. *POLYLINE (Polyline_ID, (x,y)*)*

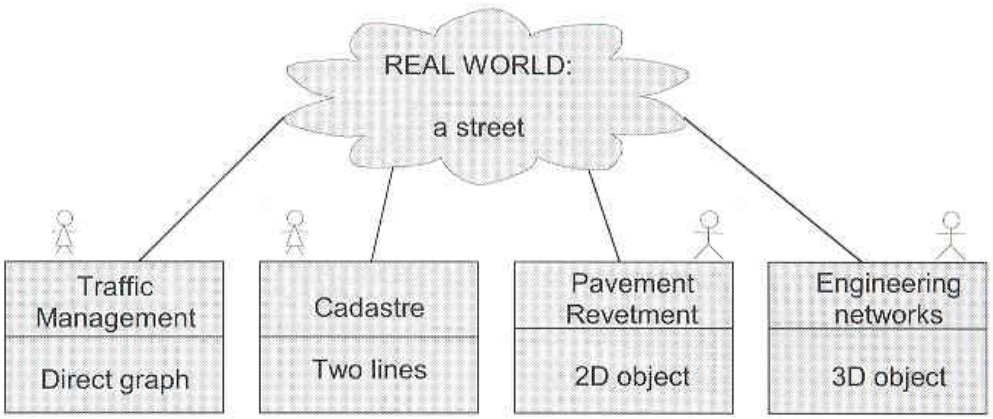

Figure 2.21 Multiple representations of a street: each user has his own model.

2.11 Multi representation

- Users can have different models for the same real features

  can be problem in the context of interoperability