MULTI-SCALE APPROACHES FOR GEODATA

Martien Molenaar
Centre for Geo-Information Processing (CGI)
Wageningen Agricultural University, The Netherlands
E-mail: MARTIEN.MOLENAAR@WETENSCH.LMK.WAU.NL
Tel: +31(0)8370-82910 Fax: +31(0)8370-84643

ISPRS Intercommission Working Group III/IV

KEY WORDS: Geo-data models, generalization, multi-scale approach, spatial objects, topology, class hierarchies, object aggregation.

ABSTRACT:

Topological object relationships in combination with object classification hierarchies appear to be fundamental in the definition of the aggregation rules for spatial objects. Such rules are essential building blocks for the construction of generalization procedures in spatial databases. A model for spatial database generalization can be formulated based on the syntax of the Formal Data Structure (FDS) as proposed in (Molenaar 1989). The syntax of the FDS will be formalised first, then database generalization procedures will be formulated with this syntax.

Four strategies will be explained for generalization, these are:

- *geometry driven generalization*, the change of geometric resolution cells determines the transition from entities at a large scale to new entities at a smaller scale,
- *class driven generalization*, spatial objects at a large scale forming a region under one thematic class are merged for representation at a smaller scale,
- *functional generalization* links objects that are considered as response units in processes defined at different scale levels,
- *structural generalization* gives a stepwise simplification of a spatial process description in an area.

These different strategies will be explained and compared. In the final discussion spatial data generalization will be presented as data transformation processes. For such a transformation we should specify which aspects of a terrain description should be invariant after generalization, this may have the effect that other aspects will not be invariant.

1. INTRODUCTION

1.1 Spatial Processes at Multi-Scale Levels

Multi-scale approaches are at present one of the focal points of the GIS research community. This is due to the rising awareness that many processes on the earth surface can only be monitored and managed if they are understood in their geographical context. Part of this context is defined by the scale range at which these processes work.

If we consider for instance the development of land use in a district, then we see that this is driven by actors at a lower aggregation level such as farmers, residents and companies. Their activities are constrained however by the socio-economic conditions and the infrastructure at regional level and by the macro economic planning at national and supra national level. An other example is the development of the natural vegetation cover of certain regions. The actual state of such a vegetation cover is defined by the co-occurrence of species that form vegetation types, which are part of eco-systems. Their development will be constrained by climatic conditions, the geologic and soil condition of the region and its hydrology. Here too we find hierarchical levels of organization.

The monitoring and management of such processes requires information at different scale levels. The research problem for the GIS community in this context is:

- to decide for each type of process which information should be handled at each scale level,
- to develop methods for transferring information between the different scale levels so that duplication of expensive data acquisition can be avoided as much as possible, and so that the consistency between data at the different scale levels can be maintained.

This second item is strongly related to the long standing research problem of map generalization, that is why it is often seen from that perspective. Researchers in this field become more and more aware of the fact, however, that multi-scale approaches in a GIS environment can be dealt with by data base generalization operations. These allow approaches that are quite different form the procedures applied in map generalization and they are more flexible.

This new research field derives it terminology and many

of its concepts from both cartography and the object oriented approaches of computer science. This mixture of the idiom of different disciplines leads often to confusion so that the concepts that are covered by the terminology become fuzzy. This confusion may send researchers in the wrong direction when they want to solve multi-scale problems. In this paper a data base perspective for multi-scale approaches will be presented, emphasising the role of topologic and semantic (hierarchical) data models.
The different concepts that play a role in the generalization of spatial data will be discussed in relation to several strategies which can be used for solving multi-scale problems.

1.2 Spatial Databases and Multi-Scale Problems

A spatial database contains data that represent in principle elementary statements about some spatial situation. These elementary statements refer to the relationships between objects and geometric data and thematic data etc. Query operations are applied to derive other statements that contain more relevant information for the user, e.g. about the state of the objects and about their mutual relationships. The semantics of the derived statements is generally of a higher level of complexity than the stored data. They should help the user to understand the structure of the mapped area, therefore they often refer to spatial relationships between the mapped objects. If the area structure should be understood at a higher abstraction level though, these derived statements could also refer to relationships among aggregated objects. The understanding of the structure of an area at several abstraction levels is strongly related to the problem of spatial generalization and multi scale representations.

Aggregation hierarchies for spatial objects can serve as basic tools for multiple representations of geo-data within the context of conceptual generalization (information abstraction) processes. These aggregation hierarchies can be based on the formal data structure (FDS) for single valued vector maps (Molenaar 1989), which combines aspects of object-oriented and topologic datamodels. Point-, line- and area objects are represented with their geometric and thematic data. Their geometric representation supports the analysis of topologic object relationships, whereas their thematic description is structured in object classes that form generalization hierarchies. These class hierarchies together with the topologic object relationships support the definition of aggregation hierarchies of objects. The classification- and aggregation hierarchies play an important role in linking the definition of spatial objects at several scale levels. Accordingly, these structures are fundamental in the definition of rules for modelling generalization of spatial information at different resolution levels. The capacity of Geographical Information Systems (GIS) to register and handle topological information in combination with object hierarchies makes them very useful tools for the automation of conceptual generalization of spatial data.

In a cartographic context, generalization can be defined as the process of abstracting the representation of geographic information when the scale of a map is changed. It is a complex process involving abstraction of thematic as well as geometric data of objects. The process usually involves two phases:

a) a conceptual generalization phase, which implies the determination of the content of a representation in the generalized situation (information abstraction), and the definition of rules how the generalized objects can be derived from the objects at lower generalization level

b) a graphical generalization phase (cartographic generalization), which implies the application of algorithms for geometric simplification of shapes and for symbolization to assure map legibility.

Information abstraction in these subprocesses is mainly determined by expert knowledge and can usually be expressed as logical rules. These rules are susceptible to be translated as database management procedures in a GIS environment (Martinez Casasnovas 1994, Richardson 1993). Regarding information abstraction, several processes are recognized: classification, association, (class) generalization and aggregation. Class generalization and aggregation are directly related to changes in the level of definition of objects when the mapping scale changes. Aggregation is the combination of elementary objects to build composite objects and will be based on two types of rules:

a rules specifying the classes of elementary objects building a composite object and

b rules specifying the geometric characteristics (such as minimum size) and topological relationships of these elementary objects (i.e. adjacency, connectivity, proximity, etc.).

The syntactic structure of a data model for handling topologic and hierarchical relationships between spatial objects will be explained in this article. Processes for database generalization will be formulated with this data model.

2. A SPATIAL DATA MODEL FOR MULTI-SCALE APPROACHES

2.1 Topologic Structures for the Representation of Spatial Objects

Entity Types for Spatial Data
The spatial structure of an area can be expressed in terms of point-, line- and area objects. Their spatial extend and their topologic relationships will be expressed by means of a set of geometric elements. (Frank ea 1986) showed that the geometric structure of a vector map can be described by means of cell complexes. For a two dimensional map these consist of 0-cells, 1-cells and 2-cells. The 0-cells and 1-cells play similar roles as respectively the nodes and edges when the geometry of the map is interpreted as a planar graph. The 2-cells can then be compared to the faces related to the planar graph through Eulers formula (Gersting 1993). The terminology of the planar graph interpretation will be used here, but their relationships will be formulated as those for cells according to the concepts presented in (Molenaar 1994). This formulation is then based on

- three geometric types: *nodes, edges and faces* represented by the symbols *n, e and f*
- three geometric object types: *point objects, line objects, area objects.*

The further developments will only use line- and area objects which will be represented by the symbol O_l and O_a. Instances of these entity types will be indicated by suffixes.

The reader will recognise that the formalization explained in this paper is to a large extent isomorphic with topologic data structures defined for GIS such as ATKIS/DLM, DGF, TIGER and DIME etc., see (Hesse 1992, Walter 1994, Marx 1990, USBUREAU 1990). This formalization will be based on the FDS described in (Molenaar 1989).

Relationships Between Nodes, Edges and Faces

The following relationships can be defined between the geometric elements of a planar graph:

- *Edge* e_i *has node* n_j *as the begin node* $\rightarrow Begin[e_i, n_j] = 1$ *otherwise* $= 0$
- *Edge* e_i *has node* n_k *as the end node* $\rightarrow End[e_i, n_k] = 1$ *otherwise* $= 0$

We will consider edges as straight line segments. Each edge will always have one face at its left hand side and on at its right hand side. These relationships will be expressed by the following functions:

- Edge e_i has face f_a at its left-hand side $\rightarrow Le[e_i, f_a] = 1$
  For any $f_b \neq f_a$ we get then $Le[e_i, f_b] = 0$
- Edge e_j has face f_a at its right-hand side $\rightarrow Ri[e_j, f_a] = 1$
  and again for $f_b \neq f_a$ we get then $Ri[e_j, f_b] = 0$

If an edge e_i has face f_r at the right hand side and face f_l at the other side then these faces are adjacent at this edge, which will be expressed by the function

$$ADJACENT[f_r, f_l \mid e_i] = 1 \text{ (and } = 0 \text{ otherwise)}$$

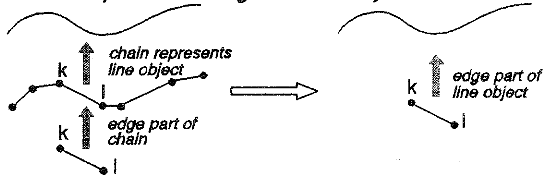
the fact that there is some edge where the faces are adjacent can then be expressed by

$$ADJACENT[f_r, f_l] = 1 \text{ (and } = 0 \text{ otherwise)}$$

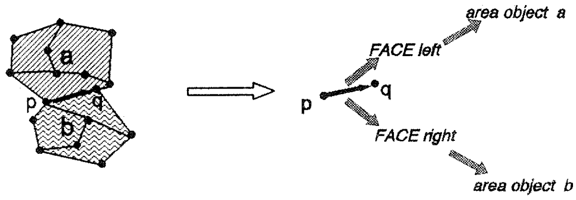
Line Objects

The geometry of a simple line object is represented by a chain of edges as in figure 1a. The fact that an edge e_p is part of the object can be established by the function $Part_{11}[e_p, O_l]$. The notation $Part_{uv}[\,]$ means that an entity with spatial dimension u is a part of an entity with dimension v. If the edge is part of the object then $Part_{11}[\,] = 1$, else it has a value $= 0$.

A line object will have a begin node $n_b = BEG(O_l)$ and an end node $n_e = END(O_l)$. These can be found through the edges of O_l, the direction of the object can then be specified by $Dir[O_l] = \{n_b, n_e\}$



a. relationship between edge and line object



b. relationship between edge, face and area object

fig. 1: Relationships between edges and objects.

Area Objects

The geometry of an simple area object is represented by one or more adjacent faces as in figure 1b. If a face f_v is part of an area object O_a this will be represented by $Part_{22}[f_v, O_a] = 1$. The set of all objects from which a face f is a part is $OA(f) = \{O \mid Part_{22}[f, O] = 1\}$

This function relates two two-dimensional entities. If there are overlapping area objects then each face might be part of several objects, but each object will also consist of one or more faces. Therefore this is a many-to-many relationship. Overlapping objects can be found through their common faces.

Now it is possible to check whether edge e_i is related through face f_v to an area object O_a. There is at most one face for which both $Le[e_i, f_v] = 1$ and $Part_{22}[f_v, O_a] = 1$. If such a face exists then the function relating the edge to the object will get the value $= 1$, in all other cases it will be $= 0$. Hence if edge e_i has area object O_a at its left-hand side then $Le[e_i, O_a] = 1$ else $= 0$. Similarly if edge e_i has area object O_a at its right-hand side then $Ri[e_i, O_a] = 1$ else $= 0$.

The combination of these two functions gives for edge e_i:

$$B[e_i, O_a] = Le[e_i, O_a] + Ri[e_i, O_a]$$

If an edge e_i is part of the boundary of O_a then only one of the functions Ri and Le is equal to 1 but not both, so for such an edge we find $B[e_i, O_a] = 1$. If e_i has O_a both at its left-hand side and at its right-hand side then $B[e_i, O_a] = 2$, in that case it is running through O_a. If $B[e_i, O_a] = 0$ there is no direct relationship between e_i and O_a.

Adjacent Area Objects

When an edge has an object O_a at its left hand side and not at its right hand side and object O_b at its right hand side and not at its left hand side then these objects are adjacent at this edge. If the objects overlap not at all, i.e. if they have no common faces and they are adjacent at least one edge, then they are adjacent which is expressed by the function $ADJACENT[O_a, O_b] = 1$ (and $= 0$

otherwise).

Line- and Area Objects

Several important relationships between a line object O_l and an area object O_a can be found by checking for each edge that is part of the line object how it is related to the area object. This will be expressed by the functions

$Le[O_l, O_a \mid e_i] = MIN(Le[e_i, O_a], Part_{11}[e_i, O_l])$
$Ri[O_l, O_a \mid e_i] = MIN(Ri[e_i, O_a], Part_{11}[e_i, O_l])$

For the relationship between a line object O_l and an area object O_a we can write

$B[O_l, O_a \mid e_i] = Le[O_l, O_a \mid e_i] + Ri[O_l, O_a \mid e_i]$

If this function has the value = 2 then the line object runs through the area object at edge e_i if the value = 1 then it is at the border and if it is = 0 then there is no relationship. The relationship between the two objects might be different at different edges.

A Hydrologic Example

For modelling hydrological systems three types of elementary objects will be defined according to (Martinez Casasnovas 1994), these are the water course lines, the drainage elements and their catchments, see figure 2. The drainage elements are gullies, each element has a catchment area from which it receives overland flow of water. Each element also receives water from upstream elements (if there are any) and it empties into a downstream element. The water flow through each element is represented by a water course line.

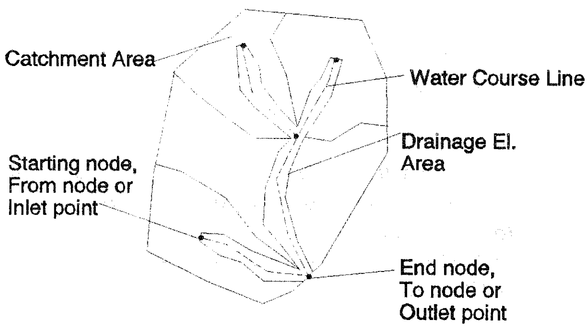


fig. 2:Elementary objects in a drainage system.

The relationship between these objects is one to one in the sense that each drainage element D_i contains exactly one water course line W_i and is embedded in exactly one subcatchment area C_i. A subcatchment area may be dissected by its drainage element, as can be seen in figure 2, but it is still considered as one subcatchment. The topologic relationships between these objects can be expressed by functions of section 2:

for water course line W_i and drainage element D_i is:
$(\forall e_k \mid Part_{11}[e_k, W_i] = 1) \Rightarrow B[W_i, D_i \mid e_k] = 2$
this will be written shortly as $B[W_i, D_i] = 2$
if $j \neq i$ *then* $B[W_i, D_j] = 0$

This means that W_i runs through D_i so that it has D_i at both sides and it is not related to any other drainage element. This is a topologic restriction due to a semantic constraint valid in the context of this hydrologic model. Another semantic constraint is

for drainage element D_i and catchment C_i is
$ADJACENT[D_i, C_i] = 1$
if $j \neq i$ *then* $ADJACENT[D_i, C_j] = 0$

so that D_i is only adjacent to C_i and to no other catchment.

Each drainage element is also connected to a downstream element and, depending on its position in the network, to one or more upstream elements. The relationship between the drainage elements can also be found through the watercourse elements. These should be directed according to the direction of the water flow, for each W_i we can find the upstream element W_h through the rule $END(W_h) = BEG(W_i)$. This relation between these water course lines will be expressed by $Upstr[W_i, W_h] = 1$, this function will have the value = 0 otherwise.

Due to the 1 to 1 relationships between W, D and C the upstream relationship can be transferred as follows

$Upstr[W_j, W_i] = Upstr[D_j, D_i] = Upstr[C_j, C_i]$

so that the order relationships between the water course lines can be translated into order relationships between the areas in which they are contained. We will assume here that the stream network structure is defined so that for each W_j with a Strahler number > 1 there are two or more upstream water lines W_i, but for each W_i there is only one downstream water line W_j.

2.2. Object Classes and Class Hierarchies

Terrain objects refer to features that appear on the surface of the earth and are interpreted in a systems environment with a thematic and geometric description. In most applications the terrain objects will be grouped in several distinct classes and a list of attributes will be connected to each class. Let Ci be a class, and let the list of its attributes be $LIST(Ci) = \{A_1, A_2, \ldots, A_n\}$ then

$LIST(Ci) \neq LIST(Cj)$ *for* $i \neq j$

i.e. these attribute lists will be different for different classes. Terrain objects inherit the attribute structure from their class, i.e. each object has a list containing a value for each class attribute, thus for member e of class C:

$LIST(e) = \{a_1, a_2, \ldots, a_n\}$
where: $a_k = A_k(e)$ *is value of* A_k *for object e*
$e \in C$
$A_k \in LIST(C)$

When two or more classes have attributes in common, then a superclass can be defined with a list containing these common attributes as "superclass-attributes" (Molenaar 1993). The original classes are subordinated to these super classes, for example, the class 'forest' is a superclass containing subclasses such as "deciduous", "evergreen", and "mixed forest". The terrain objects are then assigned to these classes.

With these observations we find the class hierarchical structure of figure 3. In literature on semantic modelling (Brodie 1984, Brodie e.a. 1984, Egenhofer e.a. 1989, Oxborow e.a. 1989) the upward links of the classification hierarchy are labelled respectively as "ISA" links. These links relate each particular object to a class and to super classes.

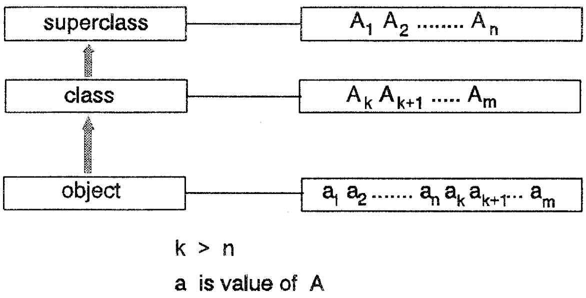


fig 3. The hierarchical relationships between objects and classes and their attributes.

It is possible to add more hierarchical levels to the structure of figure 3. At each level the classes inherit the attribute structure of their superclass at the next higher level and propagate it normally with an extension to the next lower level. At the lowest level in the hierarchy are the terrain objects, at this level the attribute structure is not extended any more, but here the inherited attributes are evaluated. In this case we find for e:

$LIST(e) = \{a_1, a_2, ..., a_n\}$
where:
$a_i = A_i(e)$ *is value of* A_i
$A_i \in LIST(C) \cup LIST(SC) \cup .....$

thus A_i is an attribute of the class *or superclass(es)* of e. If the classes at each level are disjoint so that the hierarchy has a tree structure then the terrain objects will get their attribute structure only through one inheritance line in the hierarchy, i.e. they have a unique thematic description. We will work under this assumption in this paper.

The terrain objects occur at the lowest level in the classification hierarchy. They can be seen as the elementary objects within the thematic field represented by the classification system. This implies that the decision, whether certain terrain objects should be considered as elementary or not, should always be made within the frame work of a thematic field. Objects that are considered as elementary in one thematic field are, however, not necessarily elementary in another thematic field.

2.3 Object Aggregation

Objects can be aggregated to build composite objects at several levels of complexity. These may form aggregation hierarchies which are quite distinct from classification hierarchies. An aggregation hierarchy shows how composite objects can be built from elementary objects and how these composite objects can be put together to build more complex objects and so on. In literature on semantic modelling (Brodie 1984; Brodie e.a.1984; Egenhofer e.a.1989; Oxborrow e.a.1989) the upward relationships of an aggregation hierarchy are called "PARTOF" links. These links relate a particular set of objects to a specific composite object and on to a specific more complex object and so on. For example, 'James Park is PARTOF Westminster is PARTOF London.'

For composite spatial objects the PARTOF links might be based on two types of rules involving the thematic and the geometric aspects of the elementary objects. Consequently the generic definition of a type of an aggregation should consist of the following rules (Molenaar 1993):

- *rules specifying the classes of the elementary objects building an aggregated object of this type,*
- *rules specifying the geometric and topologic relationships among these elementary objects.*

Suppose that aggregated objects of a type T should be formed. To do that we should first identify the objects O_i that could be part of such aggregates. These objects should fulfil certain criteria, which according to the two sets of rules given earlier will often be based on the thematic data of the objects. Let these criteria be expressed by a decision function

$D(O_i, T)$ = 1 *if the object fulfils the criteria*
= 0 *otherwise*

Regions can now be formed by applying two rules:

$>$ *all objects in the region satisfy the decision function for T*
$(\forall O_i \mid O_i \in R_r) \Rightarrow D(O_i, T) = 1$

$>$ *All objects that satisfy the decision function for T and that are adjacent to objects of the region belong to the region*
$(\forall O_i \mid D(O_i, T) = 1)(\exists O_j \in R_r \mid ADJACENT[O_i, O_j] = 1) \Rightarrow (O_i \in R_r)$

The second rule implies that a region can be formed when at least one object has been identified that fulfils the first rule. This object is then the seed around which the region can grow by identification of the other objects that fulfil both rules.

A region R_r can be expressed as a set of objects, i.e.:

$R_r = \{...., O_i, ...\}$

The objects of the region can be aggregated to form an aggregated or composite object O_{ar}, the suffixes express that the object is of aggregation type a and r is its identification number. The operation will be expressed by

$O_{ar} = AGGR(R_r) = AGGR(\{...., O_i, ...\})$

The fact that O_i is part of O_{ar} is expressed by

$Part_{kl}[O_i, O_{ar}] = 1$

The reverse relation expresses that the object O_{ar} consists of the region R_r, i.e. the function identifies the object that are the components of O_{ar} :

$$COMP(O_{ar}) = R_r = \{\ldots., O_i, \ldots.\} = \{ O_i \mid Part_{kl}[O_i, O_{ar}] = 1\}$$

The geometry of the aggregates can be found through the geometry of the original objects, for each geometric element we can check whether it will be part of an aggregated object of type T_a. This should be done in two steps, which will be explained for the faces of an area object O_i in relation to an aggregated area object O_{ar}. The first step evaluates the function:

$$Part_{22}[f_j, O_{ar} \mid O_i] = MIN(Part_{22}[f_j, O_i], Part_{22}[O_i, O_{ar}])$$

this function expresses whether the face is related to an aggregate through object O_i. If that is true then both functions in the expression at the right hand side of the equation will have the value = 1, and this value is assigned to the function at the left hand of the equation. If it is not the case then at least one of the functions at the left hand side will have the value = 0, so that also the function at the left hand side will get the value = 0. The second step is the evaluation of

$$Part_{22}[f_j, O_{ar}] = MAX_{Oi}(Part_{22}[f_j, O_{ar} \mid O_i])$$

If there is any object through which the face will be part of an aggregate then this function will have the value = 1, otherwise it will be = 0. If this function has been evaluated for all faces of the map then the geometry of the object O_{ar} can be found through their adjacency graph. For the edges e_i of these faces the function $B[e_i, O_a]$ can be evaluated and with this function the boundary edges can be found (i.e. $B[e,O] = 1$) and through these the topologic relationships with the other objects.

The geometry of the aggregated area object O_a can sometimes be simplified by a reduction of the number of faces. Therefor the edges e_i should be identified for which $B[e_i, O_a] = 2$, that are the interior edges. If these edges are not part of some line object so that $LO(e_i) = \varnothing$ then they do not carry any semantic information at this aggregation level and could therefor be eliminated.

The example refers to the situation where a face is related through an area object to an aggregated are object, so that all involved elements are of dimension 2. Other combinations of dimensions might occur as well, this could be the case when for example an edge is related through a line object to an aggregated area object, e.g. it is related through a river to a country.

It is possible to define aggregation types by means of their construction rules. If elementary objects are combined to form a compound object, their attribute values are often aggregated as well (as in figure 6). We will see in section 3.2 that farm yield is the sum of the yields per field, and the yield per district is the sum of farm yields. The desaggregation of such values is usually quite difficult because it can only be done if information is added to the system. An aggregation hierarchy has therefore a bottom-up character, in the sense that the elementary objects from the lowest level are combined to compose increasingly complex objects as one ascends in the hierarchy. The compound objects inherit the attribute values from the objects by which they are composed.

The PARTOF relations connect groups of objects with a certain aggregate and possibly on a higher level with another even more complex aggregate, and so on. That means that an aggregation hierarchy expresses the relationship between a specific aggregated object and its constituent parts at different levels. This is different from class hierarchies where classes at several generalization levels can be defined with their attribute structured and their intentions, but where the objects can be assigned to these classes in a later stage of a mapping process.

3. STRATEGIES FOR OBJECT GENERALIZATION

The formalism of the previous chapter helps us to express the structure of spatial datasets. This can be done in an abstracted sense, i.e. without any reference to the logic model of any implemented spatial data base. Processes applied to such datasets could also be expressed through this formalism. The four basic operations that will be used in generalization processes are:

- the *selection* of objects to be represented at the reduced scale, this selection will be based on the attribute data of the objects,
- the *elimination* from the data base of objects that should not be represented,
- the *aggregation* of area objects that should not be represented individually,
- the *reclassification* of the generalized objects.

For these four operations information about the spatial structure of the mapped area will be required. Firstly to check which relationships the objects have with their environment and what the effect of their eventual elimination will be on the spatial structure of that environment. Secondly this information is required to formulate aggregation rules for the objects that are to be merged. Once the process has been formulated one can choose how to implement it in any suitable database environment. The hydrologic example presented in sections 2.1 and 3.3 of this paper has been implemented in an Arc\Info environment, but other students of the author have made implementations of similar applications in an Oracle database, and exercises with Prolog have been made as well.

Several strategies for database generalization can be formulated with this formalism and these basic database operations .These are:

- **geometry driven generalization**: in this strategy it is the geometric information that drives the aggregation process. A clear example of this case is when the geometry of the spatial data has a raster structure. If it is then decided that the resolution of the raster will be decreased, i.e. when the cell size increases, then the original, smaller cells are merged into new larger cells. The thematic information carried by the original cells should then be transferred to the new cell.

- **class driven generalization**: in this strategy regions are identified, consisting of mutually adjacent objects belonging to the same class. These objects will then be aggregated to form larger spatial units with uniform thematic characteristics. The generalization is then driven by the thematic information of the spatial data.
- **functional generalization**: spatial objects at a low aggregation level are aggregated to form new objects at a higher level. The objects are functional units with respect to some process defined at their aggregation level, the processes at the different aggregation levels are related.
- **structural generalization**: the main aim of the process is to simplify the description of a spatial system, such as drainage networks, while keeping the overall structure intact. This to the fact that after generalization the total functioning of the same system can be understood at a less detailed level.

Each strategy has its own range of applications. Data base users should be well aware of why they are generalizing spatial data, so that they can chose which strategy is to be used. The first strategy is in most cases used when the geometric resolution of a spatial description is reduced without a clear semantic motivation. The latter three strategies, however, are semantically defined and motivated. They will be explained in some more detail now.

3.1. Class Driven Object Generalization

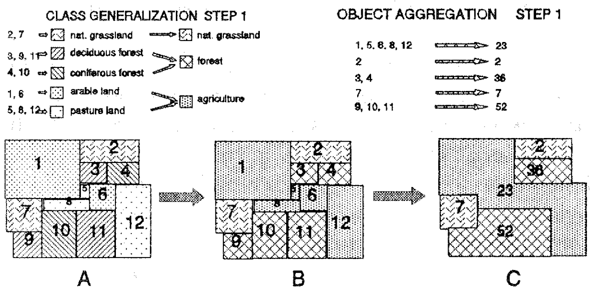


fig. 4: Class driven object aggregation.

Suppose that a database contains the situation A of figure 4, this is a detailed description of a terrain situation with agricultural fields, forrest areas and natural grasslands. This description might be too detailed for a structural analysis which should give information about the areas covered by the different major types of land use and their spatial distribution. A less detailed spatial description can then be obtained, if the original objects area aggregated to form larger spatial regions per major land use class.

Figures 4 and 5 show that this less detailed description can be obtained in two steps:

- first the objects are assigned to more general classes representing the major land use types this results in situation B of figure 4,
- then mutually adjacent objects are combined per class to form regions, this results in situation C of figure 4.

These final regions can be considered as aggregated objects. The functions $D(O,T)$ express then that objects should be aggregated per (super)class, i.e. if aggregated objects should be formed for agriculture then

if $O \in$ Agriculture $D(O, Agriculture) = 1$
else $D(O, Agriculture) = 0.$

The output of the aggregation process are regions in the sense of section 2.3. Each region is an aggregate of objects that belong to one land use class, so if R_a is an agricultural region then:

- *for all objects* $O_i \in R_a$ *is* $D(O_i, Agriculture) = 1$
- *if* $O_i \in R_a$ $\underline{and}$ $ADJACENT[O_i, O_j] = 1$ $\underline{and}$ $D(O_j, Agriculture) = 1$
  then $O_j \in R_a$

A consequence of this rule is that after the aggregation process there can be no two adjacent regions that are of the same type, i.e. that represent the same land use class.

Thematic and Geometric Resolution

The example represented in the figures 4 and 5 shows a situation where the thematic aspects of the newly aggregated objects can still be handled within the original class hierarchy. It might be that the same classes can be used as for the original objects, but the example shows a situation where it is quite clear that with each database generalization step the class hierarchy is adjusted; per step the occurring lowest level of classes is removed, only the more general classes remain, see also figures 5 and 12. That means that the thematic resolution is adjusted to the geometric resolution of the terrain description.

There might be situations where it is not necessary to jump to more general classes with each aggregation step. In those cases the new objects can be assigned to the original classes with consequence that they have the same attributes as the objects from which they have been composed. This is in fact the case if we consider the step from B to C in figure 4 in isolation. There the objects 1,5,6,8 and 12 all belong to the class "agriculture". Therefore they have the same attribute structure. They are distinct because they had different attribute values. Within this class they are aggregated to form the composite object 23, i.e.

$$O_{23} = AGGR(O_1, O_5, O_6, O_8, O_{12}).$$

This new object still belongs to the same class "agriculture" and has attributes in common with original objects. The attribute values of the original objects will then be transferred to the new object as in figure 6.

That will always have the effect that the spatial variability of the attribute values will be reduced, because after each aggregation step the attribute values that were assigned per object will then be merged into one value for a larger object. That means that the relationship between spatial and thematic resolution is not only expressed through the link between class level and aggregation level, it also

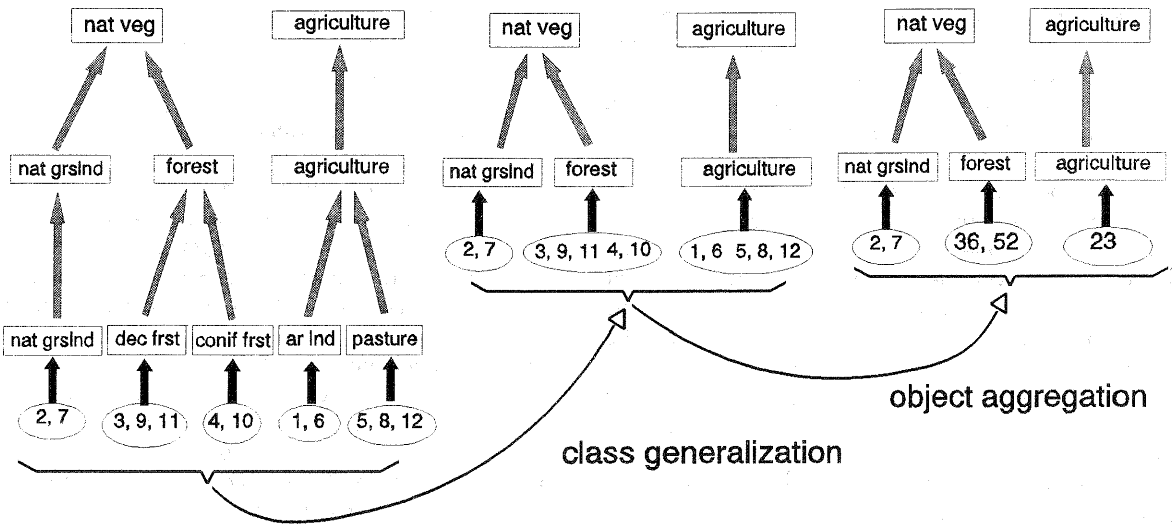


fig. 5: A diagram representing the generalization and aggregation steps of the object generalization process of figure 4.

expressed by the spatial variability of the attribute values.

When the attribute values are of the ratio scale type then the aggregated value can often be obtained by summation or by taken the average value over the objects that compose the new object, e.g.:

$$A[O_a] = \Sigma_{O_i \in COMP(O_i)} A[O_i]$$

Examples are attributes like wood volume and crop yield and population. For other attributes like vegetation cover or population density it might be that (weighted) averages should be computed.

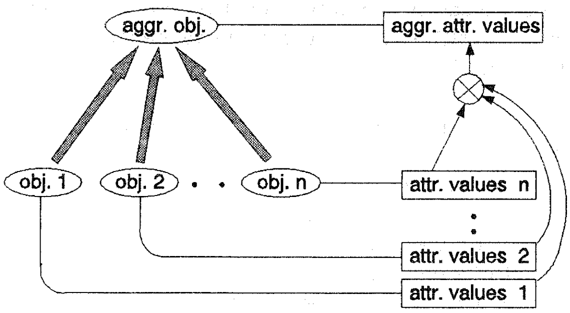


fig. 6: The aggregation of attribute values.

3.2. Functional Object Generalization

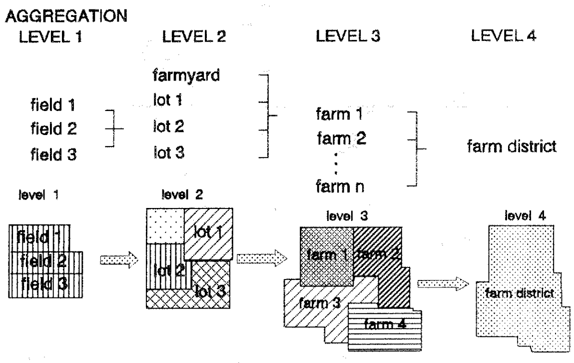


fig. 7: An example of a functional object generalization process.

It is certainly not always so that object aggregation can be done within the framework on one class hierarchy. In many cases object aggregation will imply a completely different thematic description of the objects, so that new classes should be defined. This is illustrated in figure 7 where farm yards and fields have been aggregated into farms and these in their turn into farm districts. The aggregation hierarchy has a bottom up character in the sense that starting from the elementary objects composite objects of increasing complexity are constructed in an upward direction (in figure 7 from left to right). The farm districts should only consist of farms and the farms should be mutually adjacent so that the adjacency graph (see section 4) of the farms that belong to one district is connected.

The aggregation steps in figure 7 show how the fields are considered as elementary objects at level 1. They are defined per growing season as spatial units under one crop. For the farmer they are management units, because his management operations are planned and performed per field. They are aggregated to lots which are elementary objects at level 2, i.e. these objects belong to the extensions of classes such as "arable-lot" and "grass land". These are management units at a higher level; the farmer will maintain a drainage system per lot and he will decide per growing season how to partition each lot into fields. These lots might both belong to a superclass "farm lots" in a land use data base and these again might belong to an even higher superclass "lot" which also contains the classes "forest lot" and "residence lot". The aggregation step from level 1 to 2 and the next steps to the levels 3 and 4 where we have the farms and farm districts show that after each step new objects are created. At farm level the farmer will decide whether he will be a cattle farmer or whether he will grow arable crops, in the latter case he has to decide on a rotation scheme. At district level the infrastructure and irrigation schemes will be developed. The objects at each level have their own thematic description expressed in an attribute structure that should be defined in a class hierarchy according to section 2. In this example each aggregation level requires its own classification hierarchy. This should be structured so that the generated attribute structures provide the information to support the management operations defined at the aggregation levels of the objects. The diagram of figure 8 represents the fact that a classification hierarchy should be defined per aggregation level.

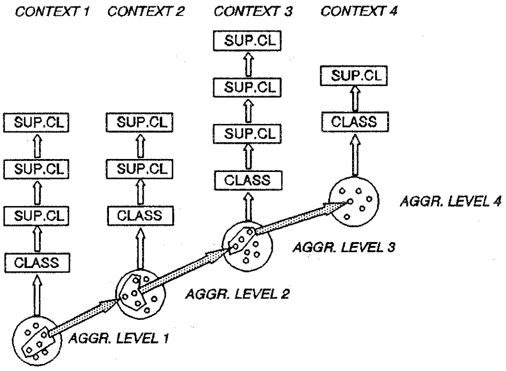


fig. 8: Classification hierarchies related to aggregation levels.

This is an example of a more general situation where objects at each aggregation level are functional units with respect to some process. In this case these were farm management processes, but we could also take examples like ecologic development, or demography and many more. Each aggregation level within such a hierarchy will have its own (sub) context within a thematic field, expressed through a class hierarchy with related attribute structures. The different (sub) contexts are related by the fact that sets of objects at one level can be aggregated to form the objects at the next higher level. There are often also relationships between various classes of the different class hierarchies related to the aggregation levels as was the case for the cover classes for the farm lots, the farm types and land use types of the farm districts at the levels 2,3 and 4 of figure 7.

There are bottom-up relationships between the objects at different levels in the sense that the state information of the lowest level objects, as contained in the attribute data, can be transferred through a process like figure 6, to give state information about the objects at higher levels. There are top-down relationships in the sense that the behaviour of lower level objects will be constrained by the information contained in the higher level objects.

3.3. Structural Object Generalization

This strategy will be explained by means of an example based on a database where the spatial description at a 1:50.000 scale, of a drainage system. The database has been structured according the FDS as in section 3, see (Martinez Casasnovas 1994). A generalization process will be executed to derive the 1:100.000 representation, so that we reduce complexity to stress spatial structure. Here the spatial structure refers to the network structure of the drainage system in relation to the subcatchments. The generalization process will keep the area of the aggregated subcatchments and the network structure of the system invariant so that the computation of overland water flows per node in the network will not be effected significantly.

The database contains geometric data and thematic data of the elements of the system, as defined in the example of section 2.1. Let the attribute ORDER contain the Strahler order number of each drainage element. These numbers in combination with the function Upstr[ ] make it possible to analyze the stream network built by the drainage elements. Through this network aggregation steps can be defined for the catchment areas. The methodology for these aggregation steps will follow to a great extend procedures defined by (Richardson 1993 and 1994).

The process starts with the identification of the drainage elements that are not mappable at the target scale, those are the elements with Strahler number = 1 with an average width $aw < Thr(eshold)$. The minimum mapping width of the drainage elements will be put at 0.75mm, that gives a threshold $Thr = 0.75mm/scale$ at terrain scale, hence $Thr = 75m$ in this case. The average width for an drainage element D_i can be computed from the $AREA_i$ of the element and the $LENGTH_i$ of its water line W_i hence
$aw_i = AREA_i/LENGTH_i$

The selection procedure applied to the drainage elements is then

\> *select the drainage elements* D_h *with* $ORDER_h > 1$
\> *select from the class with* $D_h = 1$ *the elements* D_i *with* $aw_i \geq Thr$

The set of elements that should be eliminated is then
$S = \{D_i \mid ORDER_i = 1, aw_i < Thr\}$,
their catchments should be combined with adjacent catchments to form aggregates. The elimination of the drainage elements $D_i \in S$ should consist of the following steps

\> *eliminate* W_i
\> $AGGR(D_i, C_i) = C_{DHi}$
\> *find* C_h *for which* $Upstr[C_h, C_i] = 1$
\> $AGGR(C_{DHi}, C_h) = C_{h;i}$

where the notation C_{DHi} means that the area of Di has been merged into the area of its subcatchment, the notation $C_{h;i}$ means that the area of C_{DHi} has been merged into the area of C_i. When water line W_i joins the outlet point $END(W_i)$ with only one water line W_j of a drainage element that will not be eliminated then the catchment of W_j should be merged with $C_{h;i}$

```
> find $C_j$ for which $Upstr[C_h, C_j] = 1$
if      $D_j \notin S$
and     there is no $D_{k \neq j} \notin S$ with $Upstr[C_h, C_k] = 1$
then
    > $AGGR(D_j, D_h) = D_{jh}$
    > $AGGR(C_j, C_{h;i}) = C_{jh;i}$
```

The notation D_{jh} means that D_j and D_h have been merged and $C_{jh;i}$ means that C_j and $C_{h;i}$ have been merged. When these steps have been done for each element $D_i \in S$, then new Strahler numbers can be assigned to the remaining elements according to their new position in the network. Then the selection procedure can be repeated and so on until no more elements are eliminated.

A Test Case

The drainage system represented in figure 9.a will be used as an example to demonstrate the generalization process. This figure is a schematic representation of the Romani Drainage system in the Anoia-Penedes Area in NE-Spain (Martinez Casasnovas 1994). The total area of this drainage system is 28.53 km^2, at a 1:50.000 scale it has 37 mappable drainage elements. The figure only shows the water lines W_i with their catchments, the drainage element D_i are not shown here.

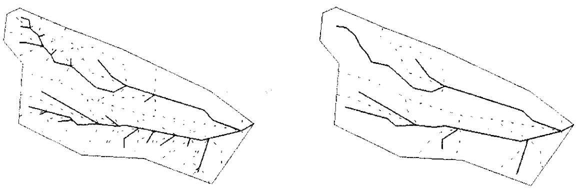


fig. 9: a) The drainage system at 1:50.000 scale representation
b) The drainage sustem at 1:100.000 scale representation

The transition from the original scale to the scale 1:100.000 will be done through the generalization procedure explained before. So first the drainage elements with Strahler number = 1 and width < 75m are eliminated, their catchments are aggregated with their downstream catchments. Then the remaining drainage elements are reclassified and the procedure is repeated until no more elements are eliminated.

The result of this procedure is shown in figure 9.b. The drainage system represented at 1:100.000 scale has only nine mappable drainage elements. Their catchments are aggregates of the catchments shown at the 1:50.000 scale. The fact that they are considered to be aggregated catchments implies that the information carried by the original catchments is now transferred to these aggregates.

The Romani drainage system has been mapped for an erosion survey. Erosion classes are estimated per catchment from the information contained in the attributes of the drainage elements. These are used to compute per catchment the drainage density in km/km^2 and the crenellation ratio in km/km^2 These data combined with the depth and the activity class of the drainage elements determine the erosion class of each catchment. When the area of the catchments are summed per erosion class we find in the original situation at 1:50.000 scale that 68% of the area is slightly eroded, 30% is moderately eroded and about 2% shows severe erosion, see figure 10.a.

After generalization aggregated catchments are formed for which the erosion classes have to be estimated. Although a large number of drainage elements have not been represented any more at the reduced scale, the information they carry has been transferred to the aggregated catchments. With these data we find now for the erosion classes that 79,5% of the area is slightly eroded, 20% is moderately eroded and 0.5% is severely eroded, see figure 10.b. These numbers deviate significantly from the original values, furthermore the spatial distribution of the occurrence of the erosion classes is quite different from the original distribution. The case is even worse if we had completely ignored the information carried by the eliminated drainage elements. Then the values would be respectively 99.5%, 0% and 0.5%.

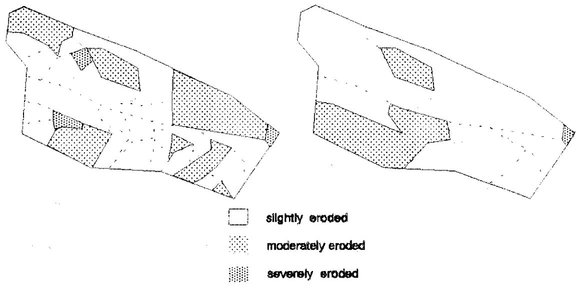


fig. 10: a) Erosion classes estimated in 1:50.000 scale representation
b) Erosion classes estimated in 1:100.000 scale representation

The structural generalization of the drainage system kept considered its constituting entities as hydrologic units. This had the effect that the computation of hydrologic processes is invariant after the generalization. The generalized network could, however, not be used to formulate reliable statements about the erosion classes of the areas in the system. That would require another generalization process where we have to specify what statements about erosion should be invariant after transformation. A class driven or a geometry driven strategy might have been more useful in this case.

4. OBJECT GENERALIZATION AND LEVELS OF SPATIAL COMPLEXITY

Chapter 3. discussed several strategies for the generalization of spatial databases. These strategies were based on the concept of spatial object aggregation in combination with class hierarchies. In the process of object aggregation the information of lower level objects is aggregated to higher level objects, but in principle the original detailed information is maintained so that it is possible to access the detailed information of the lower level objects through the aggregated objects. The result of such an aggregation process is a less detailed terrain description that may be compared to the result of a map generalization.

The output of this process could be used as the input for a following aggregation step. This has been illustrated in figure 11, that shows a process starting from the situation of figure 4.C. The regions of situation C are assigned to more general classes in situation D and the aggregated to form the larger regions of situation E.

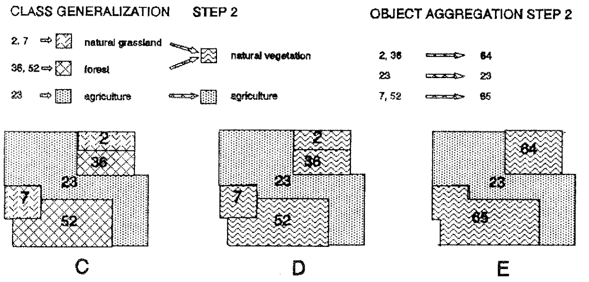


fig. 11: The second aggregation step for the objects of figure 4.C.

The class generalization and object aggregation steps of the approach of figures 4 and 11 have been represented in a different way in figure 13. This figure combines per database generalization step two steps like those of figure 4. In the first step of figure 13 the objects of the different classes are first assigned to the super classes at the next higher level in the hierarchy (compare the class generalization step of fig 4), then in the same step the objects that form a region per super class are aggregated to form a larger object (compare the object aggregation step of fig 4). This procedure is repeated in the second step of figure 13.

This figure shows that the two steps of the example of section 3.1 reduce the number of objects, that is why we rather talk of database generalization because the process generated objects with a lower spatial and thematic resolution then the original objects. Due to the fact that the original objects formed a geometric partition of the mapped area and due to the fact that generalization process made use of the topologic and hierarchical structures in which the objects had been modelled, this process resulted in a new set of objects that also formed a geometric partition of the mapped space. But the result was a terrain description of a reduced spatial complexity as is shown in the stepwise reduction of the complexity of the adjacency graphs of figure 12.

Each object is represented by a node in these graphs and the adjacency between two objects is represented by an arc. This figure gives the adjacency graphs related to each stage of a process that starts from the situation B of figure 3 where the original objects have been assigned to their

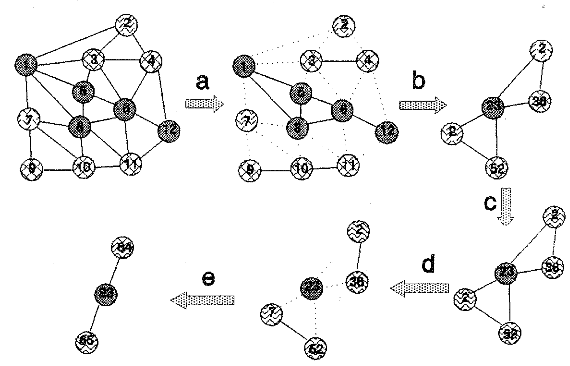


fig. 12: The adjacency graphs related to different stages of the generalization processes of figures 4 and 11.

super classes. If we follow the steps of fig. 12 then we see that:

- in step a the regions per class have been identified,
- in step b the objects in each region are aggregated to form a composite object that is represented by one node,
- these regions are after step c assigned to more general classes,
- in step d regions at this higher class level have been identified, these are composed of the objects obtained after step b,
- then finally after step e each of these regions have been aggregated again to form the objects at the higher aggregation level which is then represented by one node, this is the adjacency graph of situation E of figure 11.

The reduction of spatial complexity is one of the important aspects of generalization processes as they are known in mapping disciplines. This process has traditionally been applied in the form of map generalization to reduce the information content of a map so that a mapped area could be represented at a smaller map-scale. This process has two steps, the conceptual generalization and the graphic generalization. The conceptual generalization results in a redefinition of the mapped spatial features or objects to reduce their number for the terrain description at the smaller scale. The graphical generalization is in fact a simplification of the graphical representation of these features or objects, including such aspects as geometric simplification, object displacement, resymbolization etc.

10. CONCLUSION

When we deal with spatial database generalization in a GIS environment then this might include the graphical representation as well, but that is not necessarily so. The main aim will be a simplified terrain description, i.e. a lower spatial complexity to emphasize spatial patterns and relationships that might be difficult to find in a more detailed terrain description. That means that this process is very much related to the conceptual generalization step mentioned before the main aim of this step is to obtain a data reduction. We have seen that it can to a large extend

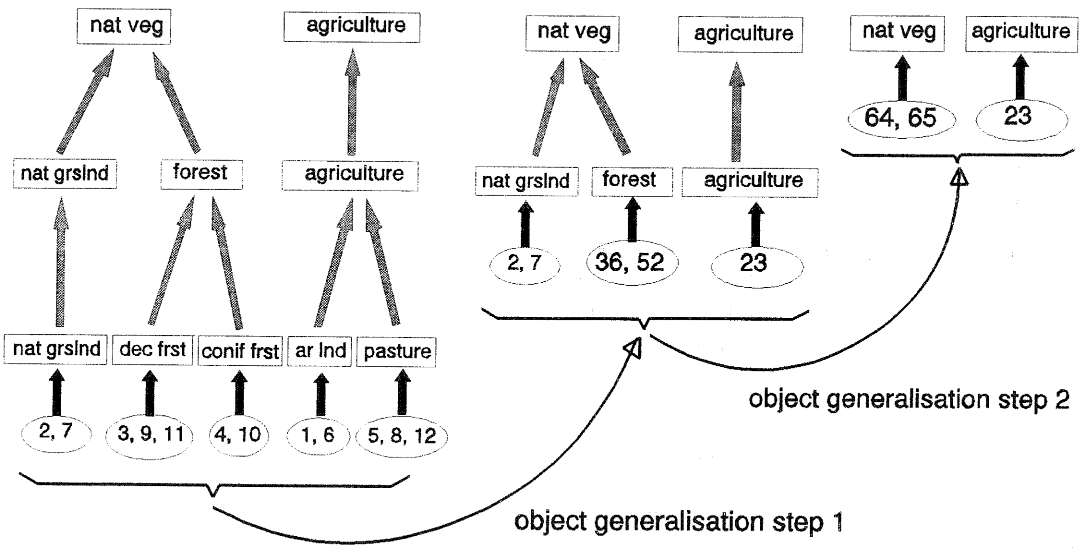


fig. 13: A diagram representing the object generalization steps of figures 4 and 11.

be defined in the form of database operations for databases that are implementations of the formal data structure (FDS) as explained in chapter 2. Such databases will be called shortly FDS-databases.

A spatial database may contain information about different aspects of a particular area, as we saw in the example of section 3.3. A generalization process may keep one aspect invariant, let us call that the primary interpretation of the database. The other aspects may be affected so that the information is not reliable after the process, we will call these the secondary interpretations of the database. If we consider generalization operations as a type of transformation of a spatial data base, then we should make explicit decisions about which aspects of the original data bases are to remain invariant, so we should decide what is to be considered as the primary interpretation of the data base. This choice will be made within some users context of the data base, i.e. the user will be interested in the correct representation of some spatial characteristics, while others may be deformed by the transformation.

A good understanding of database generalization may be useful for the design of procedures for spatial data acquisition. Information extraction from images is partly a reverse process to generalization. Generalization is a process with a stepwise data reduction, going from high resolution to low resolution. The information of the high resolution objects is merged into low resolution objects. Image interpretation can often be formulated as a process where data are produced stepwise. We can learn from generalization processes what information low resolution objects carry about their constituting high resolution objects. This knowledge may help us in image interpretation, where large image segments can be seen as low resolution objects. These should then contain thematic information in addition to the radiometric and spectral information of the image itself, to identify smaller segments that may represent high resolution objects.

REFERENCES

Brodie, M.L. (1984): *On the Development of Data Models*. in: On Conceptual Modelling (eds. Brodie, Mylopoulus, Schmidt). Springer Verlag, New York.

Brodie, M.L. & D. Ridjanovis (1984): *On the Design and Specification of Database Transactions*. in: On Conceptual Modelling (eds. Brodie, Mylopoulos, Schmidt), Springer-Verlag, New York.

Egenhofer, M.J. & A.U. Frank (1989): *Object-Oriented Modelling in GIS: Inheritance and propagation*. Auto-Carto 9, p.588.

Frank, A.U. and W. Kuhn (1986): *Cell Graphs: A Provable Correct Method for the Storage of Geometry*. Proceedings of the 2nd International Symposium on Spatial Data Handling, Seatle.

Gersting, J. L. (1992): *Mathematical structures for computer science.* Computer Science Press, New York, Third edition.

Hesse, W. and F.J. Leahy (1992): *Authorative Topographic-Cartographic Information System ATKIS.* Landes Vermessungamt Nordrhein-Wetsfalen, Bonn, 22pp.

Marx, R.W. (1990): *The TIGER system: automating the geographic structure of the United States census.* In: Introductory readings in GIS, (Peuquet, D.J. and D.F. Marble eds.), Taylor and Francis, London, pp120-141.

Martinez Casasnovas, J.A. (1994): *Hydrographic Information Abstraction for Erosion Modelling at Regional Level.* MSc. thesis, Dept of Landsurveying and Remote Sensing, Wageningen Agricultural University, Wageningen, 92pp.

Molenaar, M. (1989): *Single valued vector maps - a concept in GIS*. Geo-Informationssysteme, vol.2 no.1, pp18-26.

Molenaar, M. (1993): *Object Hierarchies and Uncertainty in GIS or Why is Standardization so Difficult*. Geo-Informations-Systeme, vol.6, No.3.

Molenaar, M. (1994): *A syntax for the representation of fuzzy spatial objects*. In: Advanced Geographic Data Modelling, Molenaar, M. and S. de Hoop eds., Netherlands Geodetic Commission, New Series, Nr.40, Delft, pp155-169.

Oxborrow, E. & Z. KEMP (1989): *An Object-Oriented Approach to the Management of Geographical data*. Conference on Managing Geographical Data and Databases, Lancaster.

Richardson, D.E. (1993): *Automatic Spatial and Thematic Generalization Using a Context Transformation Model*. (Doctoral Dissertation, Wageningen Agricultural University) R&B Publications, Ottawa, Canada.

Richardson, D.E. (1994): *Contextual Transformations and Generalizations of Remotely Sensed Imagery for Map Generation*. In: Advanced Geographic Data Modelling, Molenaar, M. and S. de Hoop eds., Netherlands Geodetic Commission, New Series, Nr. 40, Delft, pp170-178.

U.S.BUREAU OF THE CENSUS (1990): *Technical description of the DIME System*. In: Introductory readings in GIS, (Peuquet, D.J. and D.F. Marble eds.), Taylor and Francis, London, pp100-111.