Conference Paper

Map cube model - a model for multi-scale data

Author(s):
Timpf, Sabine

Publication Date:
1998

Permanent Link:
https://doi.org/10.3929/ethz-a-004364982

Rights / License:
In Copyright - Non-Commercial Use Permitted
Map Cube Model - a model for multi-scale data

Sabine Timpf
Department of Geoinformation, Technical University Vienna
Gusshausstr. 27-29, E127.1, A-1040 Vienna
Tel.: +43-1-58801-3784, Fax: +43-1-504 3535
timpf@geoinfo.tuwien.ac.at

Abstract
In this paper, we develop a model for the structure of multi-scale data derived from a map series. First we present a model of a map. We give an ontology of map elements and assign them to the objects of the model. Then we derive a model for a series of maps, connecting several of the map models with the help of tree structures. This model is called the map cube model, the horizontal axes denoting 2D space and the vertical axis denoting level of detail. A formal model of the map model and of the map cube model is given in Gofer.

Keywords: data modeling, map series, multi-scale data, multiple representations, formalization

1. Introduction
The field of multiple representations is an important research topic in Geographic Information Systems (Buttenfield and Delotto, 1989). Humans naturally use in their minds several representations, even conflicting ones, of the world around them (Tversky, 1993). Multiple representations result from seeing the world from different angles, coming from different cultures (Frank et al., 1992; Campari and Frank, 1995) and backgrounds, being in various fields of study, or domains of expertise. Geographic Information Systems (GIS) call for new modeling and reasoning methods to accommodate several perspectives.

A geographic information system serves many users. Thus, it seems only natural to store more than one representation of real-world entities in the information system. Logically there are two approaches to provide the user with several representations:

The first approach is to maintain a single database from which any other representation can be derived (Beard, 1987). This is only possible if the transformation from the stored database object to the desired output object can be accurately described, formalized and implemented (Bruegger and Frank, 1989). However, this is rarely the case. The manual cartographic generalization is a method based on this approach (Töpfer, 1974; Hake, 1975).

The second approach is to store multiple representations from a single real-world object in the database. This leads to the problem which representations to pick and how to store them appropriately in the database so that they can be accessed for the correct task (Guptill, 1989; Guptill 1990). Cartographic data vary mainly in scale and can be stored at certain ‘key scales’ in the database. Those data sets can be enlarged or reduced in scale, but only within the range of scales permissible for the data set. The ‘key scales’ depend on the tasks to be performed with the
data (Fraser, 1981). This approach relies on data structures that are hierarchical, that means, that the map objects are ordered according to their scale.

An issue discussed in this context is the problem of maintaining consistency among the objects in a database with multiple representations. Users of a GIS expect that the answers to their queries do not change with a different representation of the data. Strategies to evaluate inconsistencies and to insure consistency are the objective of research in that area (Egenhofer and Sharma, 1992; Egenhofer et al., 1994; Oosterom, 1997). This is especially interesting for the task of updating a database with multiple representation (Kilpelainen, 1994). Links between data sets at different levels of detail can be represented by scale-transition relationships if the database schemata are the same and if the data sets are consistent (Devogele and Raynal, 1995; Devogele et al., 1996).

Our aim is to derive a data structure for multi-scale cartographic data. The major problems in designing cartographic data structures are a lack of understanding of the structure of maps and of the processes that produce them (Frank, 1991). We present a model that characterizes the structure of a map and leads to a data structure suited for map series. The map model has been used to describe map data in eight case studies (Timpf, 1998). We give an ontology of map elements and assign them to the objects of the model. Then we derive a model for a series of maps, connecting several of the map models with the help of tree structures. This model is called the map cube model, the horizontal axes denoting 2D space and the vertical axis denoting the level of detail. We require that the data structures are trees or forests - this insures consistency in our data structure.

2. The Map Model

In our model, we distinguish four classes of entities in a map (Fig. 1), called map objects:

- trans-hydro network
- containers
- areas
- map elements.

From a bird’s perspective, those lines that are black and broad give a first subdivision of space. Lagrange (1994) and Bannert (1996) have proposed a division of space with the help of the road network. This idea is taken up and extended here: We subdivide a map with the hydrographic network, the railway network, and the road network.

We merge the networks of transportation and hydrography and use this trans-hydro network to create a partition of map space (Fig. 2). Buttenfield et al. (1991) have expressed a similar idea (see also Leitner (1993, 1995)).
Fig. 2: Trans-hydro networks create partitions of space at three levels of detail

Containers are those areas between the lines that the trans-hydro network creates on the map. Containers completely partition map space, if we neglect the width of the lines of the trans-hydro network. The set of containers is the dual to the trans-hydro network within the space of the map they cover.

Areas are a refinement of the container partition and thus need to be contained within a container. In contrast to the containers, areas are bounded by fences, ditches, terrain peculiarities, land-use changes, and the like. Only the borders that coincide with container borders are part of the trans-hydro network.

Elements are contained in the areas. Elements are the smallest type of entity in a map. Map elements may be houses, church symbols, other symbols, labels, terrain features, etc. They thus are mostly alike to traditional map elements, with the exception of all linear structures. The next section assigns map elements to map objects in our model.

2.1 Ontology of Map Elements

We identify the following classes of map elements in our case studies, depending on the element’s expected behavior over several levels of detail. The classification is based on geometric and semantic properties of the objects, similar to the classification by Kottik (1997). We select classes for our model and give reasons why we omit other classes.

Streets and railways: Streets and railways are included in our model as part of the transportation and hydrology network. We make a distinction between paths and dead-ends. Dead-ends are classified as map elements. Paths serve as borders of containers. The areas between paths are considered to be objects and in some cases the areas may have names like the “Quartiers” in Paris or the “Bezirke” in Vienna.

Hydrographic objects: Hydrographic objects like rivers and streams are included in our model as part of the transportation and hydrology network. Lakes and ponds are classified as areas with landuse ‘water’ or as elements embedded in an area. Sometimes the distinction between lakes and waterways cannot be clearly drawn. For example, the river Rhine flows through the lake Constance. The automatic classification of such objects is deferred to future work.

Land-use areas: Land-use areas are included in our model. The land-use class is given by the color and eventually a symbol group. We assume that a map is completely classified by land-use. The borders of land-use areas may be fences, ditches, walls, and even buildings. The symbols for area features themselves (e.g., symbols for forest or grassland) are not included in
our model. They do not structure space, but are purely graphical objects, similar to labels. A single tree symbol, however, must be included in the model.

**Buildings and Single Symbols:** We include buildings and single symbols (e.g., for pumps, churches, industrial buildings) in our model. Both types of objects are single objects embedded in land-use areas. Buildings are represented as red or black polygons. Single symbols give additional information on types of buildings (e.g., church symbol). They may also represent point objects too small to be represented, but too important to be left out (e.g., a pump symbol).

**Terrain Features:** Terrain features and terrain symbols are not included in our model. Terrain is a continuous phenomenon and it is difficult to ascribe behavior to terrain features (Buttenfield *et al.*, 1991). Terrain features do not partition space in the same sense as, for example, the trans-hydro network does. Exceptions from this rule are mountainous regions, which we do not consider here.

The cartographic generalization of terrain is a field of study that has received much attention (Weibel and Heller, 1991). The inclusion of structural components of the terrain into our model should be considered in future work. Structural components are those, which influence the cartographic generalization process. For example, two buildings on both sides of a ridge must not be merged.

**Labels:** We exclude labels from our model. Labels do not represent real-world entities on the map. They give supplementary information to the reader. The process of applying labeling to an otherwise finished map is a purely graphical process. There also exist algorithms which deal with this special aspect of map compilation as a post-processing step (Freeman, 1991; Barrault, 1995).

**Objects Appearing at Only One Scale:** We exclude objects from consideration that appear only at one scale (e.g., power lines), although they can be fitted easily into the existing model. These objects do not partition space or the map partition they create is not useful in our model.

**Three-Dimensional Objects:** Three-dimensional objects (e.g., city gates) present a problem, that could not be solved for maps and therefore cannot be solved for a multi-level database. A street runs through the old part of the city. This is not shown on the map because it passes through the old city gate and it was more important to draw the complete city walls than the continuing street. In our theory we disregard this problem, but it needs to be addressed in future work.

### 2.2 A formal Map Model

The model (Fig. 3) captures the topological and structural aspects of a map, that is, the relationships of objects on one level of detail. A map (here called Mappa to avoid confusion with a reserved word in Gofer) is determined by its name, the trans-hydro network, and a list of containers (brackets designate a list of something): $\text{Mappa} = \text{Name Tran } [\text{MCon}]$

The trans-hydro network, the container, the area, and the element are map objects:

$\text{Tran} = \text{Mob, Con} = \text{Mob, Area} = \text{Mob, Elem} = \text{Mob}$
A map-object has an identifier, a code (determining if it is a container, an area, or an element), a name, a level of detail, and a directed graph: \( \text{Mob} = \text{Mob ID Code Name Lod DGraph} \)

A container (MCon) is a map object and has a list of areas: \( \text{MCon} = \text{Con} [\text{MArea}] \)

An area (MArea) is a map object and has a list of elements: \( \text{MArea} = \text{Area} [\text{Elem}] \)

These definitions describe the structure of a map according to our model.

![Fig. 3: Model of a map](image)

### 3. The Map Cube Model

In this section, we introduce a method to describe the structural content of a topographic map series with the help of four different graphs: the trans-hydro graph, the container graph, the area graph, and the element graph. When building the graphs, we assign a vertex to each map object and an edge to each relation between map objects at different levels of detail. The resulting structure is a graph for each high level object. For example in figure 4, each graphic representation signifies a vertex. The high level object at level (b) is represented at each level down to level (d). The idea is that every map object is represented in a tree or graph, i.e., there are for every high level map object multiple low level objects, organized in decreasing levels of detail. This includes that an object may split into sub-objects. Map objects at the same depth of the tree belong to the same map.

![Fig. 4: Objects depicted with increasing graphical detail](image)

At the end of this section we combine the four graphs into a single model, the map cube model. A cube as a symbol for a database has been proposed by (Frank, 1992) in the context of user interfaces. We use the term not symbolically but literally - the two horizontal dimensions represent 2D space, the vertical dimension denotes the level of detail. We will explore all three dimensions of the cube for our map series, using the cube as a convenient image for the spatio-temporal process of zooming a map.
3.1 Trans-hydro Graph
We first subdivide map space by the trans-hydro network. This gives us a partition of map space for each map. The change of the level of detail entails that there are less and less detailed transportation networks in each map as the data get more and more abstract.

![Fig. 5: Trans-hydro Graph](image)

It is important to note, that the more abstract version of the trans-hydro network is a subset of the more detailed version. For this reason the trans-hydro graph has the structure of a filtering hierarchy (Fig. 5).

Each circle in Figure 5 represents a segment of a street. There are no railroads or waterways in this example. The colors signify that the segments belong to the same street. Two segments are filtered from level (d) to level (c), another five segments are filtered from level (c) to level (b). The red circles stand for the boundaries of the space under consideration.

3.2 Container Graph
If we zoom into the data, we see a consistent progression of the partitions towards a most detailed partition. The tree created by tracing the containers throughout the progression is called a container graph, or, more specific, a container tree (Fig. 6).

![Fig. 6: Container Graph](image)

Spaces between the edges of the network are called containers, because they contain more information, e.g., on land use, buildings, or symbols. Containers are denoted as boxes with a unique letter in the container tree. Lines between boxes signify that the container on the higher level contains the one(s) on the lower level. Or conversely, the container on the lower level is part of the container on the higher level. The container tree thus has the structure of an aggregation hierarchy. We assume in our model that the ordering is consistent, i.e. that there are no cross-links in the container graph.

Fig. 6 shows only one part of a map. Each partition of a map produces at least one tree. The collection of all trees in a map is called the container graph. If we consider the map to be the top container, then the container graph will always have the structure of a tree.
3.3 Area Graph

The graph generated by progressing through the levels of detail, looking at the areas, is called area graph (Fig. 7). This graph also represents a subdivision of space, the spaces are smaller than in the container tree, and they subdivide a single container. The partition of areas is a refinement of the partition of containers. The partition of areas is a refinement of the partition of containers. The collection of all area trees in a map is called area graph.

![Fig. 7: Area graph](image)

Fig. 7 shows an area graph consisting of three trees. Each area is represented by a box in the color and with the symbols of the original area. Lines between boxes signify the relation contains/part_of as in the container graph. The area graph has the structure of a generalization hierarchy: land-use classes are generalized. The spatial representation of the areas is aggregated.

3.4 Element Graph

The elements contained in an area are either buildings, symbols, or dead-ends. The graph created when progressing through the levels of details and linking these elements is called element graph (Fig. 8).

![Fig. 8: Element graph](image)

Fig. 8 shows the element graph for one container. It is important to notice that elements always form a graph. The graph is composed of many trees, having their root nodes at different levels. Buildings are represented as small boxes in the color of the original building in the map. Symbols are represented as symbols and dead-ends are represented as black upright rectangles. The lines between representations signify that the element represented in the next level. Ellipses and dashed lines express a special type of relation. The element graph exemplifies a structure that combines different hierarchies.

3.5 A formal Map Cube Model


196
One can imagine the structure of a digital map series as a three dimensional composition. Each map is a horizontal cut through a three-dimensional cube, the third dimension represents the level of detail, the first and second dimension represent the x and y axes of a map (Figure 9).

The cube model combines the trans-hydro, container, area, and element graph in a single model. Each of these graphs could be represented in a separate cube. All four cubes are related to each other. The trans-hydro graph corresponds to the boundaries of the containers on each level. The areas are a refined version of the containers and the elements are contained in the areas. The cube for the trans-hydro graph can be flattened to the lowest level of detail, because the other levels of detail can be calculated from the most detailed trans-hydro network.

Figure 10 shows the components of a digital map series. The same level in all graphs yields the components of a map at that level of detail. The x and y axes from Figure 9 are collapsed into a single symbolic dimension. The spatial objects of the map are represented as boxes and symbols. Figure 10 shows all the possible relationships between map objects. Each map object is related to its own type at a different level as shown on the plane. The map knowledge makes it possible to reason from one plane to the next, from left to right in the figure above, using “borders”, “contains”, and “contains” as the main relations.
A map database (MapDB) is composed of independent but interrelated elements. Trans is the superset of the trans-hydro network. Tree MCon is the container tree with edges to the embedded areas and elements. Tree Area is the area tree and [Tree Element] is the list of element trees that comprise the element graph.

```haskell
1  class MapDBS mdb where
2   createMDB :: Trans -> (Tree MCon) -> (Tree Area) ->
3          [Tree Element] -> mdb
4  getMap :: mdb -> Lod -> Name -> Mappa
5  data MapDB = MDB Trans (Tree MCon) (Tree Area) [Tree Element]
6  instance MapDBS MapDB where
7  createMDB t tm ta tob = MDB t tm ta tob
8  getMap (MDB t tm ta tob) lod n1 = TL (createTran lod n1 lod t)
9      (filter (hasLev lod) (nodes tm))

Table 1: Gofer code for a map database

The function getMap extracts a single map from the MapDB. It requires a map database, a level of detail, and a name for the map. It returns the trans-hydro network Tran and a list of map-containers MCon at the specified level of detail. The map-containers MCon include the list of map-areas MArea and their lists of elements.

3.6. Consistency in the Map Cube Model

A database needs to be consistent, because users will expect that the answers they receive do not contradict one another. We use the term consistency to describe the absence of contradictory information in the database, that is, in the map cube model. The map cube model is consistent if all graphs of the model are forests. In particular, we define the following consistency constraints:

- the edges of the trans-hydro network are classified such that they build a partition on each level of detail,
- the partition of containers at a level n is a refinement of the partition of containers at a level n-1,
- the partition of areas at a level n is a refinement of the partition of areas at a level n-1,
- the partition of areas at a level n is a refinement of the partition of containers at that level n,
- every element on level n has a representation on a lower level n+1, unless level n is the most detailed level.

We already know that an analogue map series is not made to be consistent: the level (a) map is not derived from the level (b) map but always from the most detailed map (SGK, 1987). Therefore, we expect contradictory situations in real map series from one level to the next level. These inconsistencies in the map data result in deviations from a tree or forest structure in the graphs. The deviations in the structures may be cycles or cross-edges yielding cyclic graphs, because they violate the transitivity property of trees (Timpf, 1998).

4. Results and Conclusions

The results of this paper are a formal map model and a formal map cube model for map series. We introduced four map objects, named trans-hydro network, container, area, and element. In an ontology of map elements we described which of the elements we included in our model and
how these elements are assigned to the four map objects. We have shown that a map series can be seen as a combination of four map objects: a trans-hydro graph, a container tree, an area graph, and an element graph. The tree of containers stores which container includes which areas and elements. It does not store how these areas and elements are related to areas and elements at a lower or higher level of detail. This information is encoded in the area graph and in the element graph, respectively. We have formalized and implemented the map cube model in Gofer.

Each of the four map objects is a special hierarchy. The trans-hydro graph is a hierarchy that filters levels of objects from a superset of objects. The container tree is a hierarchy that aggregates its objects to form a new level. The area tree is a hierarchy that reclassifies its objects according to a hierarchy of generic land-use classes. The element graph is a combination hierarchy that filters and generalizes objects according to graphical criteria.

The map cube model is consistent if all graphs of the model are forests. That requires that the container and area partitions at the lower levels are refinements of the partitions at the higher levels. It also requires that the area partitions at each level are refinements of the container partitions at these levels. Finally, it requires that the elements contained in one area are also contained in the area at the higher level, that means elements cannot change areas over levels.

The model does not take into account that, for example, cities are not surrounded by streets but represent a particular geographic entity that should be represented in our model. Usually, it is a problem to determine where a city ends and the only information is the administrative boundary of the city. This is the same for small villages in the countryside that are divided by a road. However, this boundary does not necessarily conform to the boundary of the built-up area. One solution could be to design a second set of containers given by the administrative boundaries. This set of containers may be very hard to relate to the first set, to the areas, or even to the trans-hydro network. Casati and Varzi (1994) have proposed to use shadows to represent and handle administrative boundaries. This would result in a set of shadow-containers. The idea deserves investigation in future work.

Acknowledgments
This work is part of my doctoral dissertation. I thank Andrew Frank for his support and help and my colleagues at TU Vienna for their patience. Special thank goes to Werner Kuhn for reading and commenting on drafts of my thesis.

References


Kilpelainen, T. “Updating multiple representation geodata bases by incremental generalization.” In Spatial Information from digital photogrammetry and computer vision in Munich, Germany, ISPRS, pp. 440-447, 1994.


