Topological Querying of Multiple Map Layers

Sylvia de Hoop[1*], Peter van Oosterom[2], and Martien Molenaar[1]

[1] Centre for Geographic Information Processing,
Wageningen Agricultural University,
P.O. Box 339, 6700 AH Wageningen, The Netherlands.
Email: dehoop@ds13.lmk.wau.nl, phone +31 8370 82643, fax +31 8370 84643.
[2] TNO Physics and Electronics Laboratory,
P.O. Box 96864, 2509 JG The Hague, The Netherlands.
Email: oosterom@fel.tno.nl, phone +31 70 3264221, fax +31 70 3280961.

Abstract. This paper first recaptures why multiple map layers are required in geographic information systems. The two main motivations are: flexibility in data modeling, and efficient processing of data. In order to make the map layer discussions clearer, we introduce two different types of map layers: a structure layer, and a thematic layer. Though the concept of a structure layer is defined in a general sense, to illustrate its practicality the organization of data in a structure layer is initially represented according to the formal data structure for *single-valued* vector maps as proposed by Molenaar. In order to develop a data model for a multi-layered system, the concept of structure layers, specified for the FDS, is extended for *multi-valued* vector maps. It turns out that the data can be modeled in various ways. After that, the topic of topological querying of multiple map layers is introduced with a few examples. Map overlay plays a central role in this process. But map overlay is a computationally expensive operation, and therefore several alternative optimization techniques are described for answering the queries efficiently. An important goal of the described multiple map layer query language is that it is a realistic approach. That is, the resulting implementation can be used in an interactive environment with real data sets: with at least several megabytes of geographic data. This is reflected by the case study presented.

1 Introduction

Most geographic information systems (GISs) use map layers to organize the geographic features. This organization may serve several purposes, among which are user convenience and software restrictions [2]. User convenience refers to the possibility to make a thematic division of data, that is, separate map layers represent specific themes. Software restrictions may require that features of a different geometric type may not be mingled in one layer. For example, the combination of point features and area features in one layer may not be allowed.

* Employed by Siemens Nixdorf Informationsystems, Munich, Germany

Modern applications demand the vertical integration of a large number of separate thematic descriptions for a single region. This requires consideration of the interaction or overlay of multiple layers of data [6]. For raster models, map overlay is straightforward. Thematic information attached to the individual grid cells is combined when two cells coincide [11, 19]. For vector models, map overlay is a more time-consuming process. This paper concerns vector models. More demanding applications involving GIS require more flexible data models for spatial information. Many of these data models accommodate the explicit representation of features, e.g. the ATKIS model [1], the DIGEST structure [10], the DLG-E data model [15], and the formal data structure (FDS) for single-valued vector maps [18]. These feature-oriented data models comprise a basic set of topological elements upon which a set of features is superimposed. This paper attempts to provide a conceptual framework for vertical integration of thematic data based on a common topological structure (the FDS).

In Section 2, the definitions of two types of map layers are given. This is done to avoid the confusion related to the concept of map layers. In the subsequent section, the concept of structure layers is elaborated on and the relationships with the formal data structure are given. The different data modeling variants are discussed in Section 4. Section 5 gives the implementation of the variants in the Postgres DBMS; it also gives some examples of the topological querying of multiple map layers. A case study on 'topographic data sets' is presented in Section 6. The paper is concluded with some remarks and an indication of future work.

2 Multiple Map Layers

A map layer should not be interpreted as a paper map, but denotes (an abstraction of) a geographic data set, i.e., a set of related geographic features. Each map layer describes a certain aspect of the modeled real world. In order to avoid confusion, a more detailed definition of map layer is required. We distinguish two types of map layers:

- The *structure layer:* a set of related features which are all based on the same topological data structure. Structure layers are independent of each other. Therefore, new layers describing a different set of features can always be added independently of the existing layers. Note that the features in a single structure layer do not necessarily belong to the same theme. That is, a structure layer may describe several themes. For example, as a road may be the boundary of a forest, the themes transportation (roads) and land use (forest) both have to be in the same topological data structure (structure layer).
- The *thematic layer:* a set of features that belong to the same theme, including all their thematic attributes or projected on some thematic attributes. A few examples of the latter are the construction year of houses, the water quality of lakes, the population of a country. If the features are part of the same

thematic layer, then they are all members of the same structure layer too. Note that within one thematic layer the class (or the geometric type) of the individual features may be different: both rivers and lakes are in the hydrography layer. With cartographic presentation techniques it is possible to display several thematic layers at the same time.

Thematic layers do not exist explicitly, but they are extracted from structure layers when required. As a structure layer can be considered separate from the other structure layers, it enables a modular approach to geographic data modeling. This is the first reason for using structure layers: it is a *natural* technique to organize the data. The usage of thematic layers is also a good modeling tool, but with more emphasis on the conceptual view instead of the physical structure (see [17]).

The second reason for using structure layers is that it is more *efficient* in terms of data storage and data manipulation. This is especially true in the case of topologically structured maps. If no topology is required (*spaghetti structure* [21]), then storing all layers together is easy. Searching for a feature that belongs to a certain theme may be time-consuming, because it has to be found among all features instead of among the features of its own theme. However, if topology is required, then combining all structure layers in one large *merged* structure layer becomes a problem when the geometry of the features is represented as a planar graph. That is, at every intersection of two lines there must be a node and both lines must be split at this node. In the merged structure layer, the area and line features are divided into many geometric segments. This fact has also been reported in [3, 4]. Features that are strongly segmented at the geometric level, have two severe drawbacks:

– inefficient storage: each 'segment' (piece of area or line feature) has to be represented; and
– inefficient manipulation: a feature has to be reconstructed from the segments, for example, in order to calculate the perimeter of an area feature.

For the data modeling and efficiency reasons described previously, structure layers should not be merged. However, this makes it impossible to directly solve topological queries that involve features that belong to different layers.

For example, one structure layer describes the hydrography of Europe, and another map layer describes the countries (Figure 1). The only connection between the two separate layers is that the metric information is given in the same coordinate system. Two example queries that may be posed are

– Through which countries does the river 'Rhine' flow?
– Which rivers flow through more than two countries?

Of course other structure layers can be defined for Europe, e.g. roads, digital elevation models, vegetation. Without loss of generality, the example is restricted to two layers. But to solve queries that deal with multiple independent structure layers, a map overlay has to be performed first. In the case of the example, a

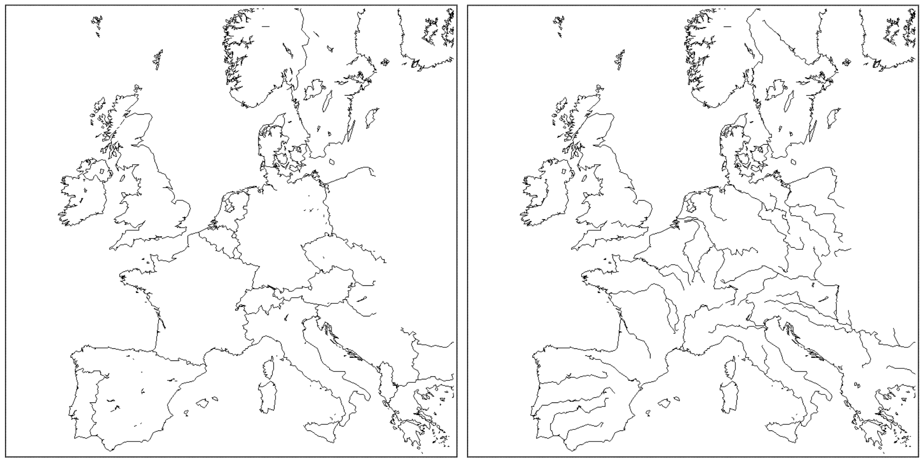

Fig. 1. Two map layers in Europe: hydrography and countries

new structure layer 'hydrography & countries' is created from the two individual layers. This new layer can be constructed, because topology can be derived from the metric information. However, the metric information has to be in exact agreement. Map overlay is a complicated procedure and is also expensive to execute from the computational point of view.

Several map overlay algorithms are described in [12, 13, 14, 16]. The first step is performed at the metric level: compute intersections (nodes) and separate the segments of a line which remain on either side of the intersection node. A reconstruction of the topology follows in the next step. The new geometric primitives inherit (or are related to) the thematic attributes of the input map layers [24]. The computationally efficient algorithms take advantage of local processing. A problem related to map overlay is the introduction of sliver polygons, these are small areas resulting from the slightly different representations of the same boundary in different map layers [12]. Solutions for this problem have been presented in [5, 25]. A formal algebraic approach has been described in [11].

If the original layers are preserved, then the merged layer is redundant, as it can be produced from the original layers. Therefore it should be considered as a temporary structure. This also avoids updating problems: updating of a feature has to be done in only the original layers, and the complex updating in the merged layer can be avoided. With the aid of this merged layer, it is now possible to efficiently answer topological queries.

3 Structure Layers and the Formal Data Structure

Several types of structure layers exist. In the situation where the features (of the different thematic layers) do not overlap, a *single-valued vector map* (SVVM) can implement this structure layer. The 'hydrography & countries' example of the previous section could be implemented using a SVVM. Subsection 3.1 will describe the formal data structure (FDS) for the SVVM. However, if features do overlap then the SVVM is not sufficient for the implementation of the structure layer. An example of such a situation is storing both soil type (thematic layer number one) and land cover (thematic layer number two) in one structure layer 'soil type & land cover'. However, a *multi-valued vector map* (MVVM) can be used to implement this structure layer; see Subsection 3.2. In Section 4 an intermediate variant on these options will be described.

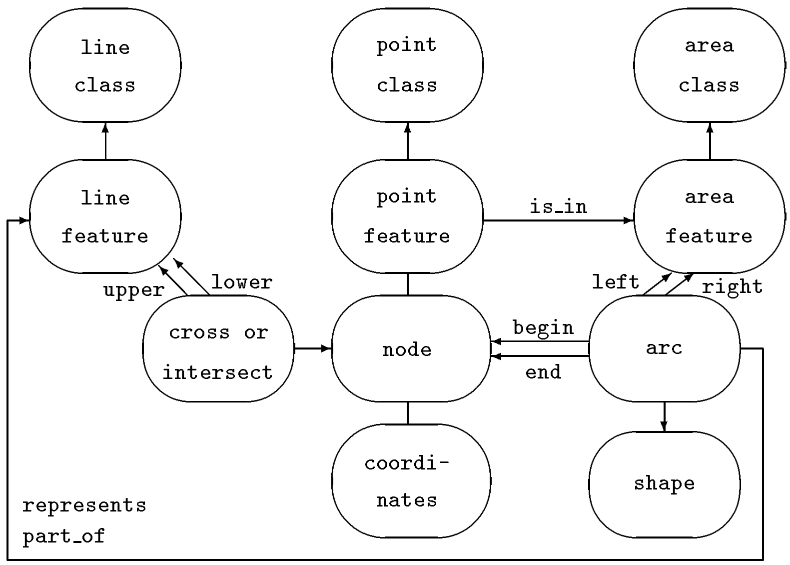

Fig. 2. The FDS for a Single-Valued Vector Map

3.1 SVVM Structure Layer

The data model presented in this section is based on the formal data structure for single-valued vector maps as proposed by Molenaar [18] (Figure 2). The arrows in Figure 2 denote many-to-one relationships in the direction of the arrow. A single-valued vector map can be interpreted as a single *integrated* structure layer. Where integrated indicates that area, line and point features may exist in the same structure layer. But, as the term single-valued denotes, features of *the same* geometric feature type do not coincide or overlap. If the organization of data in a structure layer complies with the FDS for single-valued vector maps, then this structure layer will be called a *SVVM structure layer.* A SVVM structure layer may represent several themes, e.g. 'hydrography & countries'. It is a design issue whether the two themes are represented by either two structure layers (SVVMs) or one structure layer (also a SVVM). The second solution enables efficient implementation of topological queries (related to two map layers), by using the framework of the FDS for SVVM.

Two main semantic levels can be distinguished in the FDS: a geometric level comprising metric and topology information of the geometric primitives, and a level of features where the features are described by thematic information and (indirectly) by the geometric primitives. A level of features encloses aggregation hierarchies and classification (generalization) hierarchies [20]. Spatially connected features may be put together to form complex features. In turn complex features may be put together to form more complex features, and so on. This paper considers only the lowest level in the aggregation hierarchy in which the features have a direct link to the geometry. The classification hierarchies are reduced to the most refined level of feature classifications (Figure 2). A feature can only belong to one feature class; a feature class can only contain features of one geometric feature type, i.e. point features, line features, or area features.

As Figure 2 illustrates, terrain features contain thematic data and geometric data. The former are represented by the feature classes. The latter are represented by nodes and arcs. The geometric data can be further subdivided into metric data and topological data. In the FDS, the metric information of line features and area features can be constructed partly through the topological linkages, i.e. the features are connected with the metric data (coordinates of nodes) partly through the topological data (begin node/end node references of arcs, etc.). To design an appropriate data model for a multi-layered GIS our focus is on the three main data types: thematic, metric and topological data.

Thematic data are directly feature-related, in contrast with metric and topological data which are indirectly feature-related, through nodes and arcs. Therefore, it is a good choice to represent the thematic data per feature. This organization of the thematic data is satisfactory when the GIS describes one or more themes or contains one or more structure layers.

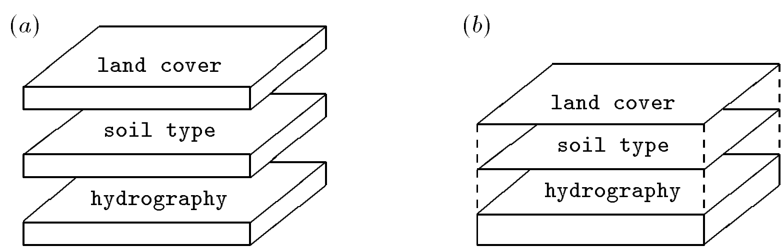

Fig. 3. Structure Layers: (*a*) multiple SVVM structure layers; (*b*) a MVVM structure layer

3.2 MVVM Structure Layer

In the situation where area features of different thematic layers do overlap, it is not possible to use the SVVM for the implementation of a common structure layer. An example of two such themes are 'land cover' and 'soil type', which are both partitions of the 2D-space. Possible classifications for 'land cover' are industry, forest, residential, agriculture, water, etc. Similarly, possible classifications for 'soil type' are sand, clay, fen, etc. Example queries involving both themes are

- Give all agricultural areas with the soil type 'sand'.
- Give all clay regions covered by 'industry' and 'agriculture'.

The representation of several themes in a GIS offers three main options for the organization of metric and topological data in a data model. Before these options are described in the next section, the concept of structure layers is explained in more detail. Beside SVVM structure layers, a *multi-valued vector map* (MVVM) can be used to implement a structure layer (Figure 3). The differences between the two types of structure layers are fundamental.

A MVVM structure layer represents several themes, but distinct from a SVVM structure layer which may represent several themes. The features of a specific theme in a MVVM structure layer may coincide or overlap with the features of another theme in the same layer. Therefore, the features in a MVVM structure layer are in general at the geometric level more segmented than in a SVVM layer. A multi-valued structure layer is the result of map overlay of two or more structure layers. In the example presented in the beginning of this subsection, one can use one MVVM for the representation of the integrated structure layer 'land cover & soil type'. This would enable the efficient implementation of the topological queries related to both themes.

In comparison with the FDS for a single-valued vector map (see Figure 2), the following applies specifically to the FDS for a multi-valued vector map: A node may represent one or more coinciding point features. An arc may represent (part of) one or more (partly) coinciding line features. With n merged structure layers, an arc has n coinciding or overlapping area features at its right hand side and n coinciding or overlapping area features at its left hand side. A point feature may lie in n area features. A line feature consists of one or more arcs. An area feature consists of at least three arcs and may additionally comprise several point features in its interior.

Except for the first one, the extensions for multi-valued vector maps denote many-to-many relationships. Note that the adjustments for multi-valued vector maps affect only the geometric level of the FDS, including the links between the geometry and the features, and not the level of features. Thus, in a multi-valued vector map the original features remain.

4 Data modeling variants

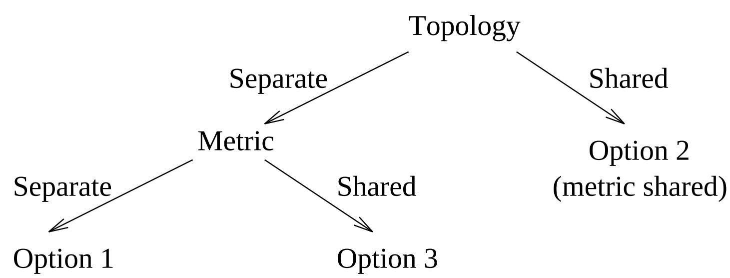

Fig. 4. The three different data models

We proceed with the three different options to organize thematic, metric, and topological data, consistent with the FDS, in case of dealing with several descriptions of the same region. As the thematic data have a direct link to the (related) features, the emphasis of the data modeling variants is on the (de)composition of the *metric* data and the *topological* data. The three options are (see Figure 4):

1. for each individual thematic layer, represent both metric and topological data per SVVM structure layer separately;
2. for two (or more) thematic layers, represent both metric and topological data in one structure layer (if possible using a SVVM, e.g. 'hydrography & countries' otherwise using a MVVM, e.g. 'land cover & soil type');
3. for two (or more) thematic layers, an intermediate structure layer with topological data organized according to a SVVM structure layer, and metric data organized according to a MVVM structure layer.

These options are similar to the proposals in [8]. The first two options represent a set of SVVM structure layers, and a MVVM structure layer covering all themes (see Figure 3), respectively. The third option combines the thematic and topological data as in a SVVM structure layer, while the metric information of all descriptions, including the coordinates of all intersection nodes (Figure 5), is gathered as in a MVVM structure layer. The word 'layer' is used in the following, to indicate that it only represents part of a geographic data set, e.g. thematic data and topological data, or thematic data and metric data. In Figure 5 the two top 'layers' comprise thematic and topological data of features that exclude each other spatially as in SVVM structure layers. Further, the bottom 'layer' comprises the metric information of the two top 'layers', including the coordinates of the intersection nodes, in accordance with a MVVM structure layer.

A fourth option would be to combine the thematic and metric data as in a SVVM layer, and gather the topological information of all themes as in a MVVM layer. This option is disregarded, because in this case the derived variant is not consistent with the FDS.

For efficient topological querying another variant on the three options can be devised; combine the thematic, metric and topological data per structure layer,

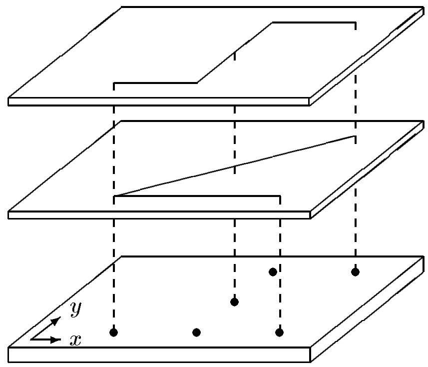

Fig. 5. The metric data, intersections included, centralized

single-valued as well as multi-valued, and *add* the topological information, encompassing all themes (Figure 6). Note that this option enlarges the geographic data set through the addition of redundant topological data. In Figure 6, the two top layers denote two SVVM structure layers, while the bottom 'layer' comprises the topological information of the planar graph obtained after map overlay.

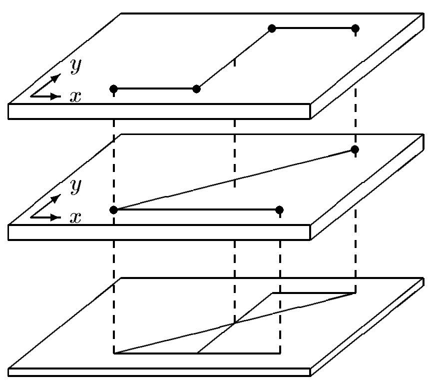

Fig. 6. The topological information, as result of map overlay, added

The above options illustrate the strength of the feature-oriented approach of the FDS. When a combination of SVVM structure layers is often used then one may consider the permanent integration of these structure layers. With regard to the FDS, the integration of structure layers may result in two different types of layers. First, the result may be a MVVM structure layer, which has already been described. Second, the result of the integration of the thematic data of different layers may be a SVVM structure layer, which means that feature classes are joined and new features are constructed and assigned to the new feature classes. The latter can be ignored, as the feature-oriented approach assures the explicit preservation of the information content of the original layers after applying map overlay.

Combinations of the options may occur in GISs. We now discuss the pros and cons of the afore-mentioned three options and the alternative, with respect to efficient topological querying. Considerations regarding redundancy and efficiency underlie the choice for a variant.

Re **1. set of SVVM structure layers** The arrangement of non-interrelated SVVM structure layers has two severe drawbacks: the same metric data may be stored more than once (redundancy), which may result in an inconsistent dataset, and topological relationships between features in different layers can only be tracked by applying map overlay which is a computationally expensive operation. Map overlay does not always have to be employed to the complete geographic region described in the database, but can in many cases be restricted to the region or features of interest. The benefit of this modeling approach is that features can be accessed directly, and topological queries restricted to one structure layer can be answered quickly. For example, as motorways may run along the border of nature reserves, both motorways and nature reserves may have the same arcs in common (see Figure 2). Therefore, to avoid redundancy these motorways and nature reserves can be stored in one SVVM structure layer. The storage of these motorways and nature reserves in separate SVVM structure layers requires map overlay when topological relationships between the motorways and nature reserves are analyzed.

Re **2. a MVVM structure layer** The gathering of both metric and topological data in one strongly segmented MVVM structure layer has the disadvantage that features have to be reconstructed from small geometric elements each time that they are accessed. Each time geographic information is added to the GIS, map overlay has to be performed. The more layers a GIS accommodates, the smaller the geometric elements will be and the larger the total volume of the data. Accuracy becomes an important factor in this modeling variant. But the advantage is that no additional calculations have to be performed when querying the different themes. Topological information of interrelated layers can be derived in a straightforward manner. The creation of a MVVM structure layer may be temporary. Applying a MVVM structure layer is especially useful when (topological) querying of interrelated layers is often requested, and the features in the different layers do not exclude each other spatially. For example, the themes ‘vegetation’ and ‘soil type’ may be combined in one MVVM structure layer ‘vegetation & soil type’. As the (interrelated) topology information is stored explicitly, topological queries related to vegetation, soil type, or both, can be answered immediately.

Re **3. an intermediate structure layer** In this layer, the thematic and topological data are combined as in a SVVM structure layer, and all metric data (including the intersections) as in a MVVM structure layer. This organization has the disadvantage that topologically interrelated features from different layers can only be traced by comparison or applying additional map overlay. The benefits of this solution are that features do not have to be reconstructed from small geometric elements and that it avoids the redundancy of metric data, as the coordinates pairs are represented only once. For example, the themes 'vegetation' and 'soil type' may also be represented in an intermediate structure layer. The query 'Give all sand regions covered by heather' cannot be answered at once, but may be answered by selecting the regions from the geographic database which comprise heather or sand. Additional matching of these regions will then provide the answer. As a relationship between vegetation and soil type can be assumed, this third option may reduce the metric data considerably. Note that the themes can also be represented in two separate SVVM structure layers.

Re **alternative** A set of SVVM structure layers extended with topological information encompassing all layers has the advantage that topological information of interrelated layers can be retrieved quickly. Furthermore, the individual features can be accessed quickly. Like the second and third option, this variant also involves the pre-calculation of the intersections of the features in all layers. The added topological data contains only the information of the topological planar graph, i.e. this layer does not contain metric data. Optionally the coordinates of the intersection nodes may be preserved. Each node and each arc in the graph is assigned an identifier. Except for the nodes which stem from the intersection of different layers, the nodes and arcs in the topological graph connect the topological data in the combined layer with the metric data in the separate structure layers. The nodes and arcs in the topological graph may contain many references to features in different layers. In addition, arcs may be the border of many area features, i.e. they may have one or more area features at their left hand side and one or more at their right. The number of topological references depends on the number of merged structure layers. Note that this possibility still requires geometric calculation at query time, when the coordinates of the intersection nodes are discarded from the topological graph. This variant involves the redundant storage of topological and metric data. Rules may be defined to guard the consistency of the data. Redundancy can be partly avoided by gathering the metric data as proposed in the second and third options. For example, one SVVM structure layer describes 'land cover' in The Netherlands, another SVVM structure layer describes 'soil type', and a third SVVM structure layer describes 'administrative boundaries'. Map overlay is applied to these layers, which yields a geographic data set (a MVVM structure layer) comprising thematic, metric, and topological data of the original layers, including the metric data of the intersections. The thematic and metric data are removed from this data set. The coordinates of the intersection nodes may be preserved. The topological data remain and are added to the original layers. With the additional topological data set, queries related to multiple structure layers, e.g. 'Give all industrial areas on sand in the province South Holland', can be answered efficiently.

5 An implementation in Postgres

¿From the discussion in the previous section, we conclude that, an optimal organization of data in a multi-layered GIS will largely depend on the intended application. Therefore, every solution will be a compromise. However, the following proposal for a data model for a multi-layered GIS, seems quite promising for topological querying a GIS and is explained in Subsection 5.1. The data model will be mapped onto the relational model with object-oriented features, as defined in the database management system Postgres [23].

In Subsection 5.2, some examples of topological queries related to multiple map layers are worked out in Postgres [23]. Open systems, such as Postgres, offer extensibility, i.e. data types, operators, functions, and index structures can be added to the system.

5.1 Relations

The relations are given in the Appendix. The following basic data types are used: `int4`, `char`, `bool` and `point`, with with obvious meanings. Variable length arrays of these data types are indicated by square brackets; e.g. `int4[]`. The set of relations is general and may be used to implement all four data models described in the previous subsection. Four groups of relations (or classes) can be identified: layers, feature classes, topological classes, and metric information. Together with a set of rules, these relations form the complete implementation.

One of the most common queries is to retrieve all features that belong to a given thematic layer (in a certain region). Though this is not a truly topological query, it can be solved with the relations given above using the `thematic_layer` class and the `tlayer` attribute of the area, line, and point features. More complicated topological queries will be described in the following subsection. The consistency and integrity rules that belong to these relations are:

1. all primary keys must be unique (by definition); note that they are used in the role of object id's;
2. all foreign keys must exist in their own table;
3. in `point_feature` (both single and multi), the `area_feature` must belong to the same structure layer as the point feature itself (has to be checked through the thematic layer);
4. in `arc` (both single and multi), the `area_feature`s and `line_feature`s must all belong to the same structure layer (single or multi-valued, respectively);
5. in case of a MVVM structure layer, the references in the classes `point_feature_multi` (for attribute `areas`) and `arc_multi` (for attributes `left_areas`, `right_areas`, and `lines`) must all be to a different thematic layer per array, because the referenced features are located in different thematic layers.

These rules can be enforced by translating them into the Postgres rule system. In this way, the consistency and integrity of the data model can be guaranteed.

5.2 Topological querying

The derivation of a set of binary topological relationships in SVVM structure layers, and extended for MVVM structure layers, has been given in [9]. It was proposed to implement the binary topological relationships as predicates. As the two parameters of a predicate can be either a constant or a variable, each predicate can represent four types of topological queries. The predicates are given intuitively meaningful names. The feature-oriented approach, followed in this paper, allows it to address the features directly, i.e. an end-user can state queries referring to the features instead of the underlying topological elements. For example, the topological query 'Through which countries does the river Rhine flow?' can be formulated in Postgres as follows

```
retrieve (area_feature.area_class)
where    Intersect(area_feature.area_class,"Rhine")
```

Note that the string `Rhine` is used as a symbolic identifier of a line feature. This query can encompass separate structure layers. In this example the selected area features are cut in two by the line feature. But depending on the interpretation of the query stated in natural language, several other predicates may be employed as well, e.g. `In`, `Permeate`, and `Meet` (see [9]). For practical use, it will in general be more suitable to group similar relationships between arbitrary features into one generic predicate. In this case, additional conditions provided by the end-user will give the specific relationship requested, e.g. in order to distinguish whether two line features touch or intersect. This may lead to a hierarchical organization of the topological relationships (predicates), as given in [22].

A similar example of a topological query, given in Section 2, is 'Which river flows through more than two countries?' By using the aggregate function `count` this query may be formulated in Postgres as

```
/* Select all countries which are intersected by rivers
   and the rivers concerned.
*/
retrieve into hr1(l.line_class, a.area_class)
from     a in area_feature, l in line_feature
where    Intersect(a.area_class, l.line_class)

/* Select every river that also intersects a second country
*/
retrieve unique (hr1a.line_class)
from     hr1a,hr1b in hr1
where    hr1a.line_class   = hr1b.line_class
and      hr1a.area_class  != hr1b.area_class
```

Topological queries can also be applied to check the semantic integrity of the data [8], for instance to sort out whether a set of windmills is located on land. The queries may address several structure layers, e.g., in the example given the specification of land may cover several structure layers. The query may be formulated as 'Give all windmills which are (erroneously) located in lakes'. Note that the topological relationship `is_in` between point features and area features is explicitly recorded in the FDS (see Sections 3 and 5.1). In Postgres, this query for a SVVM structure layer is

```
retrieve (point_feature_single.pid)
where    point_feature_single.area = area_feature.aid
and      point_feature_single.point_class = "windmill"
and      area_feature.area_class = "lake"
```

6 Theory and Practice: Topographic Data Sets

In this section, a case study is presented. The previous sections contain the theory on map layers and topology, and have shown that it is possible to model geographic data in various ways. In practice one has to decide on which way the data will be modeled. The case study in this section gives the arguments which were used in the design of a new data model for topographic base data sets of the Topographic Service in The Netherlands. The data sets concerned are related to 1:25,000 and 1:50,000 map scales and may be exchanged by using DIGEST (DIgital Geographic Exchange STandard) [10]. The data model underlying DIGEST is very similar to the FDS.

Six thematic layers are distinguished by the Topographic Service: transportation, hydrography, vegetation, buildings, administrative boundaries, and height information. One of the most important questions is 'Which themes share a topological data structure, i.e. belong to the same structure layer?' One of the first principles is that redundant data storage should be avoided. E.g., as a road may coincide with the border of a forest, they can (partly) be described by the same geometric line primitives and should therefore be in the same topological data structure. But, the integration of different themes in one structure layer may result in too many new nodes caused by the fact that unrelated features of different (related) themes intersect. Features should not be split up unnecessarily; see Section 2. However, a feature that is strongly segmented at the geometric level can still be treated as one feature. This feature-oriented approach is very useful to avoid repetition of the same thematic attributes.

It turns out that most themes are related through geometry, so they have to be in the same structure layer. However, at the feature level this is not always the case, and 'unrelated' individual features from different map layers may have to be intersected. For example, this occurs in the case of a small road through a forest. In this situation we do not want the forest to be split by the small road into two separate forests. Nodes at the intersection of the road and the boundary of the forest are introduced. For the time being, it was decided that nearly all topographic features are in the same SVVM structure layer. Except for the administrative boundaries (between municipalities, provinces, etc.) and the height information (height contours and spot heights), which have their own SVVM structure layers.

In summary, each structure layer consists of the geometric primitives points (nodes), lines (arcs), and polygons (areas). These geometric primitives are used to create features; e.g. a line feature may consist of several arcs (each running from one node to another node). Also, different types of features may be combined in creating complex features; e.g. rivers (line features) and lakes (area features) may form together a complex water feature. The thematic information is attached to these geographic features. This way of modeling requires a powerful data model, such as described for the FDS and implemented by the underlying data model for DIGEST.

It should be noted that the situation in France is quite similar to the situation in The Netherlands: in BDTopo there are two layers, one for topography and one for contour lines [7]. The same division can be noticed in Germany, where the topographic landscape model in ATKIS is divided in a 2D digital situation model and a 3D digital terrain model. The former covers topographic descriptions, like vegetation, transportation, and water(ways). The latter contains height and geomorphological information [1].

7 Conclusion

This paper addresses the confusion about the GIS concept of map layers by introducing two types of map layers: structure layer and thematic layer. These have different meanings, but they are often confused with each other in discussions about map layers. The structure layer represents the organization of geographic data, as they really reside. The thematic layer offers a view on the data. Multiple thematic layers can be extracted from a structure layer.

For the implementation of a structure layer, a SVVM can be used if the terrain features do not overlap, otherwise a MVVM can be used. A hybrid form is presented as well. Additionally, multiple structure layers can be used if the interrelationships of specific themes are unimportant. The considerations for these design decisions are presented in this paper: fragmentation of geometric primitives, redundant data storage, efficient implementation of multi-layer topological queries, flexibility in data modeling, and ease of model extensibility.

A set of Postgres relations, which implement the described data models, is given. Some multi-layer queries are given, but more research in this area is required. This can be done in combination with the case study of 'topographic data sets'. Another reseach area is the formal description of the MVVM, similar to the formal description of the SVVM.

Acknowledgments

Many valuable comments and suggestions on a preliminary version of this paper were made by Martin Feuchtwanger and Paul Strooper. We would also like to thank the Postgres Research Group (University of California at Berkeley) for making their system available.

References

1. Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV). *Amtliches Topographic-Kartographisches Informationssystem (ATKIS)*, 1988. (in German).
2. Stan Aronoff. *Geographic Information Systems: A Management Perspective*. WDL Publications, Ottawa, Canada, 1989.
3. K. Bennis, B. David, I. Morize-Quilio, J.M. Thévenin, and Y. Viémont. GéoGraph: A topological storage model for extensible GIS. In *Auto-Carto 10, Baltimore*, pages 349–367, March 1991.
4. K. Bennis, B. David, I. Quilio, and Y. Viémont. GéoTropics database support alternatives for geographic applications. In *4th International Symposium on Spatial Data Handling, Zürich, Switzerland*, pages 599–610, July 1990.
5. Nicholas R. Chrisman. Epsilon filtering – a technique for automated scale changing. In *Technical Papers of the 43rd Annual Meeting of the American Congress on Surveying and Mapping*, pages 322–331, March 1983.
6. Ian K. Crain. Extremely Large Spatial Information Systems a Quantitative Perspective. In *4th International Symposium on Spatial Data Handling, Zürich, Switzerland*, pages 632–641, July 1990.
7. Benoît David. Personnel communication through email on the topic: Map layers, June 1992.
8. Benoît David and Yann Viemont. Data structure alternatives for very large spatial databases. Technical Report ?, Institut Géographique National – France, 1991.
9. Sylvia de Hoop and Peter van Oosterom. The storage and manipulation of topology in Postgres. In *EGIS'92, Munich, Germany*, pages 1324–1336, March 1992.
10. DGIWG. DIGEST – digital geographic information – exchange standards – edition 1.1. Technical report, Defence Mapping Agency, USA, Digital Geographic Information Working Group, October 1992.
11. Claus Dorenbeck and Max J. Egenhofer. Algebraic optimization of combined overlay operations. In *Auto-Carto 10, Baltimore*, pages 296–312, March 1991.
12. James Dougenik. Whirlpool: A geometric processor for polygon coverage data. In Robert T. Aagenburg, editor, *Auto-Carto IV, Volume II*, November 1979.
13. Andrew U. Frank. Overlay processing in spatial information systems. In *Auto-Carto 8*, pages 16–31, 1987.
14. Wm. Randolph Franklin and Peter Y.F. Wu. A polygon overlay system in PROLOG. In *Auto-Carto 8*, pages 97–106, 1987.
15. Stephen C. Guptill and Robin G. Fegeas. Feature based spatial data models – The choice for global databases in the 1990's. In Helen Mounsey and Roger Tomlinson, editors, *Building Databases for Global Science*, pages 279–295. Taylor & Francis, 1988.

16. Hans-Peter Kriegel, Thomas Brinkhoff, and Ralf Schneider. The combination of spatial access methods and computational geometry in geographic database systems. In *Advances in Spatial Databases, 2nd Symposium, SSD'91, Zürich, Switzerland*, pages 3–21, August 1991.
17. Werner Kuhn. Are Displays Maps or Views? In *Auto-Carto 10, Baltimore*, pages 261–274, March 1991.
18. Martien Molenaar. Single valued vector maps – A concept in Geographic Information Systems. *Geo-Informationssysteme*, 2(1):18–26, 1989.
19. Martien Molenaar. Informatie theoretische aspecten van GIS. In *Vastgoedinformatie in de jaren negentig*, pages 67–80, June 1990. (in Dutch).
20. Martien Molenaar. Object-hierarchies. Why is data standardisation so difficult? In Kockelhoren et al., editor, *Kadaster in Perspectief*, pages 73–86. Dienst van het Kadaster en de Openbare Registers, Apeldoorn, 1991.
21. Donna J. Peuquet. A conceptual framework and comparison of spatial data models. *Cartographica*, 21(4):66–113, 1984.
22. Simon Pigot. Topological Models for 3D Spatial Information Systems. In *Auto-Carto 10, Baltimore*, pages 368–392, 1991.
23. Michael Stonebraker, Lawrence A. Rowe, and Michael Hirohama. The implementation of Postgres. *IEEE Transactions on Knowledge and Data Engineering*, 2(1):125–142, March 1990.
24. Jan W. van Roessel. Attribute propagation and line segment classification in plane-sweep overlay. In *4th International Symposium on Spatial Data Handling, Zürich*, pages 127–140, Columbus, OH, July 1990. International Geographical Union IGU.
25. Guangyu Zhang and John Tulip. An algorithm for the avoidance of sliver polygons and clusters of points in spatial overlays. In *4th International Symposium on Spatial Data Handling, Zürich, Switzerland*, pages 141–150, July 1990.

Appendix: Data Model in Postgres

```
/* PART 1: LAYERS */
  structure_layer (
      slayer_id = int4,     /* primary key */
      single_valued = bool, /* single or multi-valued layer */
      description = char[]
  )
  thematic_layer (
      tlayer_id = int4,     /* primary key */
      slayer_id = int4,     /* foreign key structure_layer */
      description = char[]
  )
/* PART 2: FEATURE CLASSES */
  area_feature (
      aid = int4,           /* primary key */
      tlayer = int4,        /* foreign key thematic_layer */
      area_class = char[]   /* thematic info */
  )
  line_feature (
      lid = int4,           /* primary key */
      tlayer = int4,        /* foreign key thematic layer */
      line_class = char[]   /* thematic info */
  )
/* PART 3A: TOPOLOGICAL CLASSES, SINGLE VALUED */
  point_feature_single (    /* also: FEATURE CLASS */
      pid = int4,           /* primary key */
      tlayer = int4,        /* foreign key thematic layer */
      point_class = char[]  /* thematic info */
      node_id = int4,       /* foreign key node */
      area = int4           /* foreign key area_feature */
  )
  arc_single (              /* primary key: from_node & to_node */
      from_node = int4,     /* foreign key node */
      to_node = int4,       /* foreign key node */
      left_area = int4,     /* foreign key area_feature */
      right_area = int4,    /* foreign key area_feature */
      line = int4           /* foreign key line_feature */
  )
/* PART 3B: TOPOLOGICAL CLASSES, MULTI VALUED */
  point_feature_multi (     /* also: FEATURE CLASS */
      pid = int4,           /* primary key */
      tlayer = int4,        /* foreign key thematic layer */
      point_class = char[]  /* thematic info */
      node_id = int4,       /* foreign key node */
      areas = int4[]        /* foreign keys area_features */
```

```
  )
  arc_multi (                /* primary key: from_node & to_node */
      from_node = int4,      /* foreign key node */
      to_node = int4,        /* foreign key node */
      left_areas = int4[],   /* foreign keys area_features */
      right_areas = int4[],  /* foreign keys area_features */
      lines = int4[]         /* foreign keys line_features */
  )
/* PART 4: METRIC INFORMATION */
  node (
      node_id = int4,        /* primary key */
      location = point       /* x,y-coordinates */
  )
```