Categorical Database Generalization

LIU Yaolin  Martin Molenaar  AI Tinghua  LIU Yanfang

1 Introduction

The main objective of database generalization is to derive a new database with different (coarser) spatial/thermodynamic/temporal resolutions from existing database(s), for a particular application.

To a large database, the efficiency in storage and access to multi-scale and multiple representation data as well as complex generalization operators need to be supported by powerful data models and data structures. Although existing data models such as Delaunay triangulation networks and formal data structure are applied to support the automated generalization, the development in this field is still at an early stage and requires much more efforts. Categorical database generalization relies on the exploitation of hierarchies which are inherent in spatial data. The past studies on categorical database emphasized on the geometric and visualization aspects only. There is lack of method research on supporting data model, statistics analysis, semantic analysis and spatial analysis, which play a key role in the operations in generalization.

2 Semantic supporting model for database generalization

The contents of categorical database are always closely related with a taxonomic system, such as soil database with soil taxonomy system, landuse database with landuse taxonomic system, etc. The taxonomic system is used in the real world to establish hierarchies of classes.
that permit us to understand, as fully as possible, the relationships among entities or between entities and properties which are responsible for their character in the real world. Some concepts must be defined before discussing the semantic model.

2.1 Class (object type) and object

In this paper, the class and object type have the same meaning. A class or object type is defined by the attributes shared. A class or object type determines a set of attributes to form its attribute structure. An object is an instance of an object type or class. The attribute structure of objects is determined by the class to which they belong, so that each object has an attribute structure list containing one value for every attribute of its class. An object inherits the attribute structure of a class to which it belongs. The thematic description of an object can now be specified by its class (which specifies the attributes for the object) together with the list of attribute values.

2.2 Classification hierarchy and aggregation hierarchy

Classification and aggregation hierarchies play a key role in defining conceptual data model of categorical database since the object types in the conceptual data model are meaningful within a certain classification and aggregation hierarchy. Before a categorical database can be built, the classification structure must be chosen and the aggregation structure must be specified.

A classification hierarchy is used in the context of database expressed as an object type hierarchy that represents the levels of object specificity. Furthermore, a classification hierarchy as an abstraction type organizes the levels of both objects and object type definition and reflects the abstract level of objects in the database. For the categorical database which is always related to a taxonomic system in a certain application filed, a classification hierarchy is derived from the taxonomic system. In this sense, we can say that the object types in the classification hierarchy correspond to the classes, in the taxonomic system. The super object types and sub object types in the classification hierarchy correspond to the super classes and sub classes, respectively. The objects of the object type correspond to the entities of the class. This system can be easily transformed into the classification hierarchy in the database (Fig. 1).

The aggregation hierarchy is expressed in the way that a higher-order object is organized by lower-order object types that belong to different classification hierarchy in the sense that the aggregation hierarchy can be derived from the classification hierarchy with a certain application purpose.

Even if we can specify the relations between higher-order object types and lower order ones to build an aggregation hierarchy, the specifying relations are normally based on the classification hierarchy. In function, the classification hierarchy will help us to find the objects we need, because it has sorted and categorized them in the categorical database. Once we find them, the aggregation hierarchy tells us how to put them together meaningfully, for example, an aggregation of river and road object types into a transportation network develops a significantly different definition from the individual definitions of river classification and road classification (Fig. 2).

For the categories database, the object types, attribute structure of each object type and relationships among the object types in
3 Transformation model of database

The database generalization can be considered as the transformation of the content of a spatial database from high resolution to a lower resolution terrain representation. In fact, the database generalization is a transformation from one existing state of a database at certain detail level to a new state at less detail on the basis of the application and users requirements (Fig. 3). The state of database (SDB) can be specified by a five tuple,

\[ SDB = \{ M, O, R, C, P \} \]

where SDB is the state of database at a certain detail level, \( M \) is the set of conceptual data models, \( M = \{m_i\} \); \( O \) is the set of objects, \( O = \{o_1, o_2, o_3, \ldots, o_n\} \); \( R \) is the set of relationships among the objects, \( R = \{r \mid r \in O \times O\} \); \( C \) is a set of conditions or constraints for transformation; \( P \) is a set of operators.

In the categorical database transformation, several aspects must be taken into account.

3.1 Conceptual data model transformation

The conceptual data model is an abstraction of the real world of the interest field. It consists of object types and relationships among the object types in the context of database. It plays an important role in database transformation. It determines what object types and which instances of these object types should be contained in the database. It determines the degree of detail of the target database and the contents of database as well. Database is the instance of conceptual data model. In a sense, we can say that the database generalization is the transformation from one data model of an existing database to another data model of a generalized database on the basis of the application purpose. This means that if the user introduces a new conceptual data model, it will lead to a database transformation from high resolution to lower resolution. For the categorical database, the conceptual data model of a database has a close relationship with the classification and aggregation hierarchy and taxonomic system in application field. The classification and aggregation hierarchies play an im-
important roles for linking the definition of spatial objects at several scale levels and definition of spatial object types at several scale levels. These hierarchies play an essential role for defining the conceptual data model of the categorical database. Before transforming an existing database to a new database, a new data model associated the new database must be defined.

3.2 Object and relations among object transformations

In a sense, the concrete content of a database consists of objects and the relations among the objects. An object is an instance of an object type. When an existing data model is transformed into a new data model, a set of object types which form an existing data model will be replaced by a set of object types of new data model. When object types are changed, their instances will be changed as well. This replacement will result in the transformation of objects and relations in the existing database. Some objects in the existing database maybe disappear or form new objects in the new database. The change in objects will induce the change in relations among the objects. For example, two spatially adjacent objects with different attribute in the existing database will be merged to form an homogeneous object, if their attribute becomes the same. The adjacent relation between the objects will disappear after being merged. Transforming the objects and relations among the objects from the existing database to a new database is the concrete content of the database transformation. The transformations of objects are involved in the geometric and thematic properties of the objects. The transformations of relations include those of spatial and semantic relations.

The object and relation transformation deal mainly with the preservation of typical shapes (on the object level) or with the preservation of the spatial patterns and alignments of objects, whereas the conceptual data model transformations deal with the preservation of the logical context of objects and degree of detail (on the object type level). The data model transformation controls the objects and relations transformation.

3.3 Transformation conditions or constraints

The transformation of a database is conditional. Database transformations are controlled by a set of conditions (i.e. application purpose and requirements), called constraints. These constraints govern, or guide the transformation process of database. Before transformation, the constraints which will influence the transformation must be identified and classified. They are application-dependent. The constraints of the transformation are involved in the aspects of conceptual data model, spatial and semantic properties of objects and relations among the objects. The constraints have the properties of multi-level.

3.4 Transformation operations

To finish transforming database from high resolution to low resolution, some operations are needed to be done. Some of researchers have discussed the operations in the map generalization and the database generalization from different points of view. Operations for the object type level and operations for the object level are quite different. The object types are at higher level than the objects in the database in the sense that the former is at decision level and the latter at the operational level in the process of building the new database. The operations which will be introduced in the database generalization should reflect this characteristic.

4 Spatial supporting data structure

Categorical database generalization is a process of analyzing, understanding and decision-making on geographical objects. Not only the information about the object, but also the information about the environment (such as neighbors) that constraints the behavior of the object are all required in the process. On the other
hand, the process also requires the geometric data of the object and the attribute data of the object. In this case, it needs adequate data structure which can be used to represent the objects and detect the natures of spatial relationships and interaction among geographical objects in the database to support the generalization decision-making and the operation implementation.

Formal Data Structure (FDS) for single valued vector maps is an object-oriented topological (conceptual) data model, which combines the aspects of object-oriented and topologic data model. Point, line and area objects are represented with their geometric and thematic factors. Their geometric representation contains the information about the topologic object relationships, whereas their thematic description is structured in object classes that may form generalization hierarchies. Such class hierarchies in combination with the topologic object relationships of FDS support the definition of aggregation hierarchies of objects. These classification and aggregation hierarchies play important roles in linking the definition of spatial objects at several scale levels. An important property of constrained Delaunay triangulation (CDT) is the adjacent relationship between two points connected by a Delaunay edge. Each Delaunay edge in a Delaunay triangle represents the topology between two points. The triangulation should be so equilateral that the unexpected effect of long elongated edges can be minimized. CDT can be used for defining adjacent relations among connected or disconnected objects, conflict detection and displace of spatial objects, finding nearest neighboring object to a given object in generalization. CDT can also be used to measure the spatial relations such as disjoint relation, distance relation and direction relation. For the categorical database generalization, CDT is very useful to analyze and measure local spatial relationship but not to organize the whole data set since a simple area object will consist of a lot of triangles that will lead to date redundant too much and also lead to the difficulty of semantic analysis among objects. CDT may be generated dynamically and locally at a certain step of a generalization process.

In the real world, the concept of adjacency may also include the adjacent relations between those area objects that are geometrically disconnected from each other, between line objects, between point objects and between objects of different geometric description types. FDS can not support these kinds of adjacent and inclusion relations. In order to make full use of the FDS and CDT, the advantages of FDS and CDT are combined into the development of a data model which is the dynamic integration of FDS and CDT in the database generalization transformation process (Fig. 4).

![Fig. 4 Data model for database generalization](image-url)
5 Semantic similarity analysis model

Whether the two adjacent objects or an adjacent group of objects can be merged or aggregated depends on the attributes of the two objects which are the same or similar or not. If their attributes are the same or similar or higher than the threshold, they can be merged or aggregated. Otherwise not. The closeness among objects and object types can be described by the similarity. The similarity is application-dependent. Classification and aggregation hierarchy are an ordered structure as discussed above. These hierarchies can reflect the similarity between object types at the same level or at different levels. In a sense, the similarity will control and guide the database transformation operations.

5.1 Hierarchic semantic similarity matrix

For a hierarchical structure, a semantic similarity matrix can be defined on the basis of properties of the hierarchical structure, as shown in Fig. 5. The semantic similarity matrix represents the similarity between object types, between object and object type, and between object types.

![Fig. 5 Matrix of similarity](image)

In Fig. 5, obj1, obj2 etc. denote different elementary objects; Sub-type1, sub-type2 etc. denote different elementary object types; sup-type1, sup-type2 etc. denote the composite object type; $s_{ij}$ denotes the similarity value between matrix elements.

The larger the value of an element in the matrix is, the closer or more similar the objects or object types that the element links are. The matrix is symmetric and reflexive and has the property that $s_{ij} = s_{ji}$ in the matrix. Here, $s_{ij}$ is a value between 0 and 1.

This matrix has the characters of hierarchy and shows the similarity between the objects of the same level and between the objects of the different levels. This will provide potential possibility for the objects in the same level and between different levels to be merged or aggregated. This also means that the objects of different levels can be kept in a spatial database. The similarity matrix will be used as a look-up table for guiding or governing the aggregation process of spatial objects in semantics for a certain application.

The value of element $s_{ij}$ in the matrix can be given by expert knowledge or the calculation based on the aggregation hierarchy and the classification hierarchy and the requirements and purposes of the database generalization.

5.2 Computing model of similarity

On the basis of the set theory, Tversky (1977) defined a similarity measure in terms of matching process based on the normalization of Tversky's model and the set theory. This measure produces a similarity value that is not only the result of the common, but also the result of the different characteristics between objects.

A natural approach to compare the degree of similarity between object types is to determine the distance from these object types to the immediate super object types that sub-
sumes them, that is, their least upper bound in a partially ordered set. A computational model that assesses the similarity among objects and object types bases on some definitions, concepts and hierarchical structures. The similarity model for the categorical database is proposed as:

\[
s_u(c_i, c_j) = \begin{cases} 
    l + a(c_i, c_j)d_{i} + (1 - a(c_i, c_j))d_{j}, & (c_i \text{ and } c_j \in \text{the same sub-tree}) \\
    \beta + a(c_i, c_j)d_{i} + (1 - a(c_i, c_j))d_{j}, & (c_i \text{ and } c_j \notin \text{the same sub-tree}) 
\end{cases}
\]

(1)

where \( l \) is the shortest distance (number of the link edges) from the immediate super object type that subsumes \( c_i \) and \( c_j \) to the top of a hierarchy; \( d_{i} \) is the shortest distance (number of the link edges) from the immediate super object type that subsumes \( c_i \) and \( c_j \) to \( c_i \); \( d_{j} \) is the shortest distance (number of the link edges) from the immediate super object type that subsumes \( c_i \) and \( c_j \) to \( c_j \); \( a \) is a function of the distance (number of the link edge) between the object types \( c_i \) and \( c_j \) and the top of a hierarchy; \( \beta \) is the correlation degree among different sub-trees (A or B), such as the similarity among agriculture land use, forest land use and building land use, and its value can be given by experts according to application requirement.

Eq. (1) is suitable for two cases. One is for two given objects or object types belonging to the same sub tree such as \( A \) in Fig. 6, and the other is for two given objects or objects belonging to different sub trees as \( A \) and \( B \). For the first case, the model uses two types of distances to define the common and different properties between the given objects or object types; one is the distance between the given objects or object types and the immediate super object types that subsumes them which reflects different properties between two given objects or object types; the other is the distance between the immediate super object types that subsumes two given objects or object types and the top of hierarchical structure which reflects the common properties of two given object or object types. For the second case, the distance between the immediate super object types that subsumes two given objects or object types and the top of hierarchical structure will be zero since the two given objects or object types belong to different sub trees. So this distance will be replaced by the correlation value between two sub trees in Eq. (1).

In Eq. (1), \( a(c_i, c_j) \) can be expressed as a function of the distance of the \( d_{i} \) and \( d_{j} \). In order to get the final values of \( a \), the function is defined as:

\[
a(c_i, c_j) = \begin{cases} 
    \frac{d_{i}}{d_{i} + d_{j}}, & (d_{i} \leq d_{j}) \\
    \frac{d_{j}}{d_{i} + d_{j}}, & (d_{i} > d_{j}) 
\end{cases}
\]

(2)

where \( d_{i} \) is the shortest distance (number of the link edges) from the immediate super object type that subsumes \( c_i \) and \( c_j \) to \( c_i \); \( d_{j} \) is the shortest distance (number of the link edges) from the immediate super object type that subsumes \( c_i \) and \( c_j \) to \( c_j \).

This similarity function yields values between 0 and 1. The extreme value 1 represents the case when everything is common between two entity classes, whereas the value 0 occurs when everything is different between two entity classes.

6 Transformation units and their creation

Transformation unit is a group of objects or object types, in which there are adjacent relations among these objects or semantic relations among these object types and there ex-
ists at least one conflicted object or conflicted object type. The conflicted object is the seed through which we can find a set of objects which have adjacent relations with the object to form a transformation unit. The transformation unit proposed in this paper is an important process unit as many generalization problems need to be solved by considering a subset of related objects as a whole. In a sense, it is a basic analysis that processing and decision-making unit plays an important role in the database transformation. Four types of transformation units are considered in this study, each of which requires a different generalization solution.

1) Creating transformation unit based on thematic constraints
After the data model associated with the original database varies into a new data model with a target database, the first step we should take is to detect thematic conflict of adjacent objects. This group of adjacent objects with thematic conflict will be built as a transformation unit.

2) Creating transformation unit based on geometric constraints
The objects that violate geometric constraints and their spatial adjacent (connected) neighbors will be built as different transformation units.

3) Creating transformation unit based on spatial relation constraints
A set of adjacent (disconnected) objects in which the distance (space) among them is less than the threshold will be built as a transformation unit.

4) Creating transformation unit based on geometric and spatial relation constraints
A set of adjacent (connected or disconnected) objects in which there exist geometric and spatial relation conflicts (violated geometric and spatial relation constraints) will be built as a transformation unit.

7 Case study
Fig. 7 shows a part of subset land use database, in which the original land use data are organized according to the concept of FDS in the thematic and geometric aspects. After that, the data are processed to meet the requirement of triangulation network for analyzing the spatial relations and detect the conflicted objects. After establishing a new classification hierarchy and implementing changing attributes of the objects in a database, the transformation unit based on thematic meaning must be formed. Before the aggregation of objects, the similarity evaluation among objects within a transformation unit must be done. Fig. 7 shows the thematic conflicted objects with dash line. The thematic conflicted objects will be aggregated with the object which has the same object type at next higher level or the semantic similarity value with the conflicted object higher than the threshold.

A number of unimportant small objects or their areas less than the area threshold will be detected first (Fig. 8) to build the transformation. These conflicted objects will be as seeds to form transformation units. For a conflicted object an object will be found by using the triangulation network. If the distance between these objects is less than the distance threshold, they will form a transformation unit, and later they will be aggregated. If these objects do not exist, a conflicted object with its adjacent objects will form a transformation unit. The semantic similarity among the objects will be evaluated by the model within a transformation unit. The object having the highest semantic similarity with the conflicted object within the transformation unit will be selected and aggregated with the conflicted object. Fig. 9 gives examples of the transformation units. Fig. 10 shows the result of a subset of land use database generalization.

8 Conclusions
This paper introduces the contents of categorical database generalization, integrated
and enhanced spatial supporting data structure and semantic evaluation model for database generalization. This data model and integrated data structure and the evaluation model in combination with classification and aggregation hierarchy play a very important role in data description, organization, spatial analysis, decision-making and implementation of database generalization transformation. The transformation units provide the information about processing objects, which makes spatial analysis and process more efficient and effective.

References

1. Molenaar M (1996) Multi-scale approach for geodata. *International Archives of Photogrammetry and Remote Sensing*, XXXI(B3):542-554

2. Molenaar M (1996) The role of topologic and hierarchical spatial object models in database generalization. In: Molenaar M ed. Methods for the Generalization of Geo-database. Delft, Netherlands Geodetic Commission.

3. Molenaar M, Martinez C G A (1997) A formalism for the structural description of vector maps and its use for multi-scale representations: a hydrographic example. *Cartographica*, 33: 55-63

4. Molenaar M, Richardson D E (1994) Object hierarchies for linking aggregation levels in GIS. Symposium of ISPRS Comm. IV, Athens.

(Continued on Page 26)