Geo-spatial Information Science
Vol.4, No.3, p.42~49 Sept.2001

FRAMEWORKS FOR GENERALIZATION CONSTRAINTS AND OPERATIONS BASED ON OBJECT-ORIENTED DATA STRUCTURE IN DATABASE GENERALIZATION

LIU Yaolin
Martien Molenaar
AI Tinghua
LIU Yanfang

KEY WORDS constraints; database generalization; operation; spatial object; hierarchy

ABSTRACT The constraints and the operations play an important role in database generalization. They guide and govern database generalization. The constraints are translation of the required conditions that should take into account not only the objects and relationships among objects but also spatial data schema (classification and aggregation hierarchy) associated with the final existing database. The operations perform the actions of generalization in support of data reduction in the database. The constraints in database generalization are still lack of research. There is still the lack of frameworks to express the constraints and the operations on the basis of object-oriented data structure in database generalization. This paper focuses on the frameworks for generalization operations and constraints on the basis of object-oriented data structure in database generalization. The constraints as the attributes of the object and the operations as the methods of the object can be encapsulated in classes. They have the inheritance and polymorphism property. So the framework of the constraints and the operations which are based on object-oriented data structure can be easily understood and implemented. The constraint and the operations based on object-oriented database are proposed based on object-oriented database. The frameworks for generalization operations, constraints and relations among objects based on object-oriented data structure in database generalization are designed. The categorical database generalization is concentrated on in this paper.

1 Introduction

Database generalization can be considered as the transformation of the content of a spatial database from high resolution (with more detail) to a lower resolution (with less detail) terrain representation (Molenaar, 1996). In other words, this transformation is deemed as changing state of database from one state to another.

Project supported by the International Institute for Aerospace Survey and Earth Science, Ministry of Education and the State Bureau of Surveying and Mapping.
LIU Yaolin, Professor, Ph. D candidate, School of Resource and Environment Science, Wuhan University, 129 Luoyu Road, Wuhan 430079, China
E-mail: lyl@hp01. wtusm. edu. cn

The contents of OO-database mainly consist of objects and relations among objects (implicit or explicit) from a user's point of view. The database is the instance of the data schema. Changing the data schema of database will modify the content of database. This means that database generalization transforms an existing database only if the user has introduced a new data schema which will lead to a new database. In a sense, the transformation of a database mainly is the transformation of the objects, relationships among the objects and data schema associated with database through a set of operations. In other words, the transformation of the data schema, object characteristics, relations among objects are the main content in database generalization. Nevertheless, this transformation is application-dependent. It is controlled by requirements (conditions) and purpose of application. These conditions are called constraints borrowed notion from map generalization. Some authors have discussed constraints in map generalization, such as Robinson, *et al.*, 1984; Brassel and Weibel, 1988; Weibel and Dutton, 1998; Beat and Weibel, 1999; Beard, 1991; Ruas, 1998, 1999; etc. In the context of map generalization, a constraint can be defined as a design specification to which the solutions to a generalization problem should adhere (Weibel and Dutton, 1998).

For constraints in database generalization there is still lack of research on database generalization. We can consider constraints as the specification of conditions of data schema, objects and relationship which govern the process of database generalization transformation. The constraints as transformation conditions play a key role in the process of database generalization transformation. During the process of generalization, constraints may be used: ① to identify areas that have to be generalized, for evaluating the quantity and severity of constraint violations, ② to guide the choice of operators according to constraints priorities, ③ to control the effect of an algorithm by detecting constraint violations on objects after each transformation (Ruas, 1996).

A database is assumed to exist. The data in database are organized by Formal Data Structure (FDS) for a single valued vector map which is an object-oriented topological (conceptual) data structure. It consists of: ① three feature types, namely point feature, line feature, area feature, classified according to the geometric description of spatial object, ② four geometric data types (geometric primitives), including coordinates, node, arc and shape, the definition of which is based on planar-graph theory at node-arc level, ③ a set of links between geometric data types (g-g links), and a set of links between geometric data types and feature (g-f links). It supports a number of elementary topological relationships, including area-area, line-line, point-point, area-line, area-point, and line-point relationships (for more detail of FDS see Molenaar, 1989, 1991).

2 Constraints to spatial database generalization

Constraints which are pertinent to database generalization must be identified, and one must also identify connections between constraints. The goal of this section is to identify constraints that are applied to the generalization of spatial database. Three types of constraints are defined in database generalization according to the requirements of database generalization and components of database. They are data schema constraints, object constraints and relation constraints. These three types of constraints are discussed more in detail later. We concentrate our study only on constraints related to data model, objects and relation among objects though the list is not exhaustive.

The object and relation constraints deal mainly with the preservation of typical shapes (on the objects level) or with the preservation of patterns and alignments (relationships among objects) if multiple objects are involved. Data schema constraints deal with the preservation of the logical context of objects and degree of detail.

2.1 Constraints to spatial data schema (model)

Spaital data schema (model) plays a key role in database generalization, in a sense that database generalization can be considered as the transformation from the existing spatial data schema with detail to new spatial data schema with less detail. The spatial data schema determines the generalized result. Lanza an La Barbera (1993) proved, in this respect, that a water network can be generalized differentially, according to the criteria of hierarchy model used. In the categorical database, the spatial data schema could include classification hierarchy and aggregation hierarchy. Both hierarchies reflect a certain aspect of data abstraction. The classification and aggregation hierarchies play an important role in linking the definition of spatial objects at several scale levels (Molenaar, 1996; Peng, 1997; Peng and Tempfli, 1997; Richardson, 1993; Smaalen, 1996).

2.1.1 Classification hierarchy constraints

Object types and super-types can be organized into a hierarchical structure called classification hierarchy (Smith, 1977; Thompson, 1989; Hughes, 1991; Molenaar, 1993). The classification hierarchy is not only very important in building a new database or reorganizing data in the existing database, but also for deriving aggregatin hierarchy and constraints. The top-down relationship of classification hierarchy is called IS-A links. This relationship makes it possible to transform a more complex model to a less complex one. Classification hierarchy may be some classification systems, such as soil classification or land use classification, or may be derived classification system from existing database according to application requirement, such as hydrological network classification through Horton (1945), Strahler (1964), and Shreve (1966). The classification hierarchy can be constructed by attribute structure, function, and order of objects etc. Because object types at different levels in a classificaton hierarchy correspond to data of different complexity, in this sense, specifying an (elementary) object type implies, to a certain extent, to determine the abstraction/complexity level of a geo-spatial model. Changing the object types of an existing data schema (model) to the ones at a higher level in the same hierarchy would mean transforming the data schema (model) from a lower abstraction level to a higher abstraction level (Peng, 1997). The constraints of classification hierarchy may include:

① hierarchical structure associated with existing database,

② In the hierarchy lower levels corresponding to lower abstraction levels and resulting in more complex data, including both thematic and spatial aspects, whereas the higher levels corresponding to higher abstraction levels and leading to less complex data,

③ level in which an object type is located in its associated classification hierarchy corresponding to the degree of abstraction,

④ level in which the associated domain of an attribute of an object type is located in its associated classification hierarchy corresponding to the degree of abstraction,

⑤ number of elementary object types,

⑥ number of attributes contained in an object type,

⑦ attribute structure of each object types,

⑧ similarity evaluation among objects, object types, sub-object types, and super object types,

⑨ one object only belonging to a class and a super class, and one class having many objects,

⑩ specifying a specific object type and its constituents at different levels.

2.1.2 Aggregation hierarchy constraints

Another important structure is the aggregation hierarchy (Hughes, 1991; Molenaar, 1993). This structure shows how aggregated objects can be built from elementary objects that belong to different classes and how these composite objects can be put together to build more complex objects and so on (Molenaar, 1998). In this article, a higher-order object type in the hierarchy is called composite-type, whereas an object type that is a part of the composite type is called component-object. Accordingly, an instance of the composite-type is referred to as an composite-object, and an instance of the component-type is regarded as a component-object. A composite-type can be the component-type of another (super) composite-type. This implies that replacing the component-types in a model with their composite-type will result in transforming the model from a lower abstraction level to a higher abstraction level (Peng, 1997).

The upward relationship of an aggregation hierarchy is called "PART-OF" links. These links relate a particular set of objects to specific composite object and on to a specific and more complex object and so on. Such class hierarchies in combination with the topologic object relationship of the FDS (Molenaar, 1989) support the definition of aggregation hierarchies of objects (Molenaar, 1998). The constraints for aggregation hierarchy may include:

① hierarchical structure associated with existing database,

② composite-types in the hierarchy corresponding to higher abstraction levels and resulting in less compliex data,

③ component-types corresponding to lower abstraction levels and resulting in more complex data,

④ level in which an object type is located in its associated aggregation hierarchy corresponding to the degree of abstraction,

⑤ level in which the associated domain of an attribute of an object type is located in its associated aggregation hierarchy corresponding to the degree of abstraction,

⑥ number of attributes contained in an object type,

⑦ number of composite-type,

⑧ number of component-type,

⑨ similarity among component object, composite-object, component object type and composite objects,

⑩ specializations (rules) specifying the component types of the component objects building an composite object of this type,

⑪ specializations (rules) specifying the geometric and topologic relationships among these objects,

⑫ specifying a specific composite object and its constituents at different levels.

2.2 Constraints to objects

A spatial object contains both thematic and geometric information and is represented in a database by means of an object identifier.

Three types of constraints can be identified on the basis of the characterisitcs of spatial objects. They are thematic constraints, geometric (spatial) constraints and temporal constraints. Temporal constraints and related aspects are not discussed in this paper.

2.2.1 Thematic constraints

Thematic constraints are a specification that indicates the thematic abstraction level of the objects in a database. It includes:

① the same geo-phenomena should be described using the same thematic resolution through out the entire data model,

② no object has common boundaries with other objects having the same object type, (If the case occurs, the separating boundary is dropped.)

③ connected (adjacent) objects belonging to the different object type may be aggregated,

④ adjacent (disconnected or connected) objects having the same object type or different object types may be merged,

⑤ adjacent (disconnected or connected) objects having the different object types may be aggregated,

⑥ the area of eliminated object should be added to the area of the object which has the highest similarity with eliminated object among its neighboring objects, or be distributed to the area of each neighboring object if its neighboring objects have almost the same similarity with it,

⑦ respect the size ratio for each class relative to the total area.

These aspects and the number of object types that a database contains determine the thematic constraints of the database. Thematic constraints may be ranked by nominal, oder, interval and ratio, but can not be measured.

2.2.2 Geometric constraints

The geometric (spatial) constraints of objects in a database mainly deal with aspects of the size, width and distribute structure of objects. It mainly meets the requirements of application. In other words, they are application-dependent. It comprises:

① geometric description type,

② minimum object size (minimum size for area objects, or minimum length for line objects),

③ minimum object's detail that a database can contain,

④ objects which are too narrow,

⑤ minimum space between objects,

⑥ respecting the global shape and angularity of objects,

⑦ respecting typical shpaes and angularity of objects of each object types,

⑧ respecting the given size distribution of objects for each objects types,

⑨ preserving typical alignments and patterns of objects within a group of objects,

⑩ avoiding merging two objects that the distance between two objects is less than the minimum space, but the type is different.

These two aspects of constraints of spatial objects are applied partly to objects of an object type, partly to the entire database, and may take different values for different object types in the same database.

2.3 Constraints to relations among objects

Two types of constraints can be identified among spatial objects. There are spatial relations and semantic relations between objects. Spatial relations are classified into topological relations, direction relations and distance relations. Eenghofer and Herring (1991), for instance, provided a mathematical framework for the definition of topological relations, while Papadias and Sellis (1993) provided framework for the definition of topological relations defing direction relations using representative points. The different application has different requirements to the relations. Depending on the application domain, some spatial relations may be more important than others. As for database generalization in which the data are organized by FDS for single valued vector map, the authors only use topological relationships (connectivity, adjacent, inclusion) and proximity relations for database generalization.

To generalize a set of objects, it was necessary to have a great deal of information on spatial and semantic relations of objects. According to Kate Beard, spatial relations between objects present constraints for each generalization operation. Spatial and semantic relations constraint governs the process of database transformation.

2.3.1 Topological relations constraints

Topological relationships should be preserved. This is the general principle of generalization after generalization in the database. They constrain the behaviors of objects in spatial aspects. The topological relationships determine the neighbors of one object. Topological relations constraints include:

① topological constraints deal with basic topological relationships like connectivity, adjacency and containment, which should be maintained when generalizing data;

② self-intersection and overlapping objects do not exist and cannot be introduced with generalization;

③ an object must not move across the boundary of another object;

④ an object must not overlap with another object;

⑤ introduction of illogical neighborhood relations (e.g. house in a lake) is avoided;

⑥ separation of object when deleting parts of it is avoided;

⑦ introduction of self-intersection of object outlines is avoided.

2.3.2 Proximity relation constraints

Proximity constraints define the relative position constraints of a set of disjoint objects. To a certain degree, proximity relation can reflects distance relationships among objects. Proximity relation within and between two objects can guide the transform of an object into that of another. Proximity relationships may be deseribed by means of structure such as a local Delaunay triangulation to form an object triangulation which would describe the proximity relationship within area. Both interior dimensions of objects and the spacing between them are relationships which may need to be priserved or in other ways constrained. Proximity constraints include:

① avoiding merging two adjacent objects of the same type, the distance between which is larger than the minimum space,

② preserving the relative location of one object in relation to other ones after generalization,

③ objects of the different object type having proximity relationship may be aggregated if the distance between them is less than the minimum space,

④ objects of the same object type having proximity relationship may be amalgamated if the distance between them is less than the minimum space,

⑤ objects of the same object type or the different object type having proximity relationship may be merged if the distance between them is less than the minimum space.

2.3.3 Semantic relation constraints

Semantic relations are also of essential importance to reduce the number of objects in an object type. Semantic relation between two objects limits objects behavior in semantic aspect in the database.

These constraints depend on the database specification. Semantic constraints should be used for indicators. It contains:

① objects with a sub object type having IS-A relationship may be amalgamated to form the object of the higher object type;

② objects with a component object type having PART-OF relationship may be aggregated to the composite object of the composite object type;

③ objects with a composite object type having

PART-OF relationship may be aggregated to the composite object of the hhigher composite object type;

④ a set of small objects having the same similarity in semantic aspects may be merged to a larger object;

⑤ an object having a specific function should be maintained.

2.4 Representation of constraints

Constraint violations allow us to indicate where an action should be preformed. If we want to use constraint violates as triggers, we need to represent them in the database.

Constraints violations related to an object can be represented by means of attributes at the object level (e. g. too small area, too detailed line) with either a flag or a quantitative value which describes the severity of the violation.

Constraints violations relate to a class of objects can be represented at the class level or by means of a specific attribute which is an indicator that should be consulted during the process (Ruas, 1998).

3 Operations in database generalization

Reducing the number of objects in database generalization is the main task (Molenaar, 1998; Weibel, 1995). We should bear this in mind when we define the operations in database generalization. In a database of objects, all objects can be independently selected for display. So a reduction in the number of objects can be accomplished by simply selecting a desired set from the database or replacing several objects with a new object. We can define six operations in database transformation on the basis of geometric, semantic relations and constraints.

1) Selection: a selection operation that selects object types and objects of object types for target database. The purpose is that the objects are retained selectively according to the requirements of application and constraints.

2) Aggregation: an aggregation operation that merges two or more adjacent (disconnected or connected) objects of the different object type to form an object of super object type performed through "PART-OF" relationship. The values of some of the attributes may need to be modified after aggregating two objects or more into a larger one.

3) Merge: a merge operation that merges two or more connective or disconnected but adjacent objects of the same type or different object type to a larger object of the prevailing object type among those objects in size or importance. The values of some of the attributes may need to be modified after merging two objects or more into a larger one.

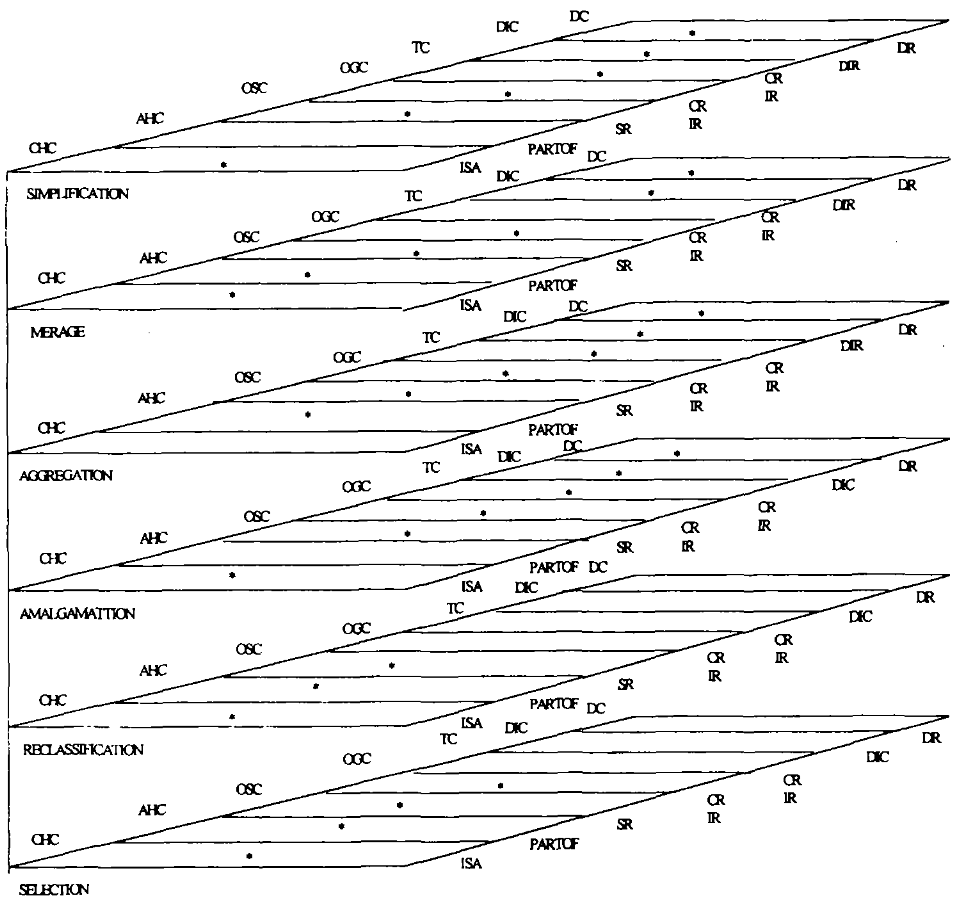
4) Amalgamation (generalization): an amalgamation operation that amalgamates two or more adjacent (disconnected or connected) objects of the same object type to form a larger object of super object type performed "IS-A" relationship between object types. The values of some of the attributes may need to be modified after generalizing two objects or more into a larger one.

5) Reclassification : an reclassification operation that aims at creating instances of a new objects type using objects of another object type, of which one of the attributes defines the theme of the new object type (Pang, 1997).

6) Simplification: a simplification operation that reduces the number of attributes of an existing object types but leaving the theme unchanged and simplifies the dimension of object of unchanged object type.

4 Modeling constraints and operations

Most constraints do not work independently, they are contextually related and affect one another. A constraint violation occurs when an object or a set of objects do not respect a constraint. For example, an object whose area is too small to violate a size constraint, two objects which are too close to violate a proximity constraint. An object violating the constraint requires some operation, but several operations are possible to resolve. The choice of operation depends on application purpose, characteristics of the object, the spatial neighborhood of objects, or other variables of significance to the user. Operations are applied to a database to correct, or preserve conditions specified by constraints. In the context of this approach, the function of operation must be clearly defined to anticipate or predict how they will interact with constraints. Fig. 1 shows the diagram among operations, constraints and relations. The priority order from bottom to top are decreased on the basis of semantic first. Each layer represents one operation. The sign represents the constraints and relations for an operation in a layer.



CHC Classification Hierarchy Constraints
AHC Aggregation Hierarchy Constraints
OSC Object Thematic Constraints
OGC Object Geometric constraints
TC Topological Constraints
PC Proximity Constraints

ISA IS-A
PARTOF PART-OF
SR Semantic Relation
CR Connectivity Relation
IR Inclusion Relation
PR Proximity relation

Fig. 1 Operations, constraints and relations

5 Conclusion

The constraints as transformation conditions play a key role in the process of database generalization transformation. Constraints can be used to identify conflict area, guide the choice of operations and trigger operation.

The three types of constraints, data schema, object and relations based on object-oriented database have been proposed in database generalization. Constraints can be specified interactively by users and varied to reflect different objectives or purpose. All three types of constraints are application-dependent. This will make the database generalization process very flexible/adaptive, and decision-making is on the basis of geographic meaning and not simply the geometry of an object.

In order to reduce the detail of database and make function of each operation individual, and avoid related operations, six kinds of operations are developed based on constraints and relations. These operations meets requirement of databse generalization process. The modeling of operations and constraints and relations are identified.

References

1 Beard K, Mackaness W. Generalization operations and supporting structures. Auto-Carto 10, 1991. 29～42

2 Beard K. Constraints on rule formation. In: Buttenfield B P, McMaster R B, eds. Map Generalization: Making Rules for Knowledge Representation. Essex: Longman House, 1991. 121～135

3 Beat P, Weibel R. Integrating vector and raster-based techniques in categorical map generalization. Third ICA Workshop on Progress in Automated Map Generalization, Ottawa, 1999

4 Egenhofer M J, Herring J R. A mathematical frame work for the definition of topological relationships. In: Brassel K, Kishimoto H, eds. Proceedings of 4th International Symposium on spatial Data Handling. Zurich, 1990. 803～813

5 Hughes J G. Objects-oriented databases. New York: Prentice Hall, 1991

6 Laurini R, Thompson D. Fundamentals of spatial information systems. London: Academic Press, 1993

7 McMaster R B. Conceptual frameworks for geographical knowledge. In: Buttenfield B P, McMaster R B. Essex: Longman House, 1991. 21～39

8 McMaster R, Barnett L. A spatial-object level organization of transformations for cartographic generalization. Auto-Carto 11, Minnesota, 1993. 386～393

9 Molenaar M. Terrain objects, data structures and query spaces, In: Schilcher M, ed. Geo-Informatik. Munchen: Siemens-Nixdorf Informationssysteme A. G, 1991. 53～70

10 Molenaar M. Object hierarchies and uncertainty in GIS or why is standardisation so difficult. *Geoinformations Systeme*, 1993, 6(3): 22～28

11 Molenaar M. Multi-scale approaches for geo-data. *International Archives of Photogrammetry and Remote Sensing*, 1996, 31(*B*3): 542～554

12 Molenaar M. Single valued·vector maps —— a concept in GIS. *Geoinformations System*, 1989(2): 18～26

13 Molenaar M. the role of topologic and hierarchical spatial object models in database generalization. In: Molenaar M, eds. Methods for the Generalization of Geodatabses. Delft: Netherlands Geodetic Commission, New Series, 1996, 43: 13～36

14 Molenaar M. An introduction to the theory of spatial object modelling for GIS. London: Taylor & Francis, 1998

15 Muller J C. Generalization of spatial data bases. In: Maguire D J, Goodchild M F, Rhind D W, eds. New York: Longman Scientific & Technical Ltd, 1991, 1: 457～475

16 Peng W. Automated generalization in GIS: [Ph. D Thesis]. The Netherlands: Wageningen Agricultural University (and ITC), 1997

17 Peuquet D. A conceptual framework and comparison of spatial data models. *Cartographica*, 1984, 21, (4): 66～113

18 Pilouk M. Integrated modelling for 3D GIS: [Ph. D Thesis]. The Netherlands: Wageningen Agricultural university, 1996

19 Richardson D E. Automated spatial and thematic generalization using a context transformation model: [Ph. D Thesis]. Canada: R & B Publications, 1993

20 Ruas A, Lagrange J P. Data and knowledge modeling for generalization. In: Muller J C, Lagrange J P, Weibel R, eds. GIS and Generalization, 1995. 73～86

21 Ruas A, Plzanet C. Strategies for automated generalization. *SDH* 96, 1996, 6A: 1～15

22 Ruas A. OO-Constraint modeling to automate urban generalization process. *SDH* 98, 1998. 225～235

23 Shea K S, McMaster R B. Cartographic generalization in a digital environment when and how to generalize. *Auto-Carto* 9, 1989. 56～65

24 Smith J M, Smith D C P. Database abstractions: aggregation and generalization. ACM Transactions on Database System, 1977, (2): 105～133

25 Thompson P J. Data with semanues: data models and data management. Van Nostrand Reinhold, New York, 1989

26 Van Smaalen J W N. A hierarchic rule mdoel for geographic information abstraction. *SDH* 96, 1996. 4B: 31～41

27 Ware M J, Jones, Chris B. A spatial model for detecting and resolving conflict caused by scale reduction. The 7th International Symposium on Spatial Data Handling, Delft, The Netherlands, 1996, 9A. 15～23

28 Weibel R. Three essential building blocks for automated generalization. In: Muller J G, Lagrange J P, Weibel R, eds. GIS and Generalization, 1995. 56～69

29 Weibel R, Dutton G H. Constraints-based automated map generalization. The 8th International Symposium on Spatial Data Handling, Vancourver, BC, 1998. 214～224

30 Weibel R. A typology of constraints to line simplification. *SDH* 96, 1996, 9A: 1～13

31 Worboys M F. GIS — A computing perspective. London: Taylor & Francis, 1995