Frameworks for Generalization Constraints and Operations Based on Object-Oriented Data Structure in Database Generalization

Liu Yaolin [1], Martien Molenaar [2], Ai Tinghua [1]
[1] School of land information and Cartography
Wuhan University
430072,Wuhan,P.R China
E-mail:yaolin@itc.nl,
[2] Department of Geo-Informatics
International Institute for Aerospace Survey and Earth Scienc
7500 AA, Enschede,The Netherlands
E-mail: Molenaar@itc.nl

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 the existing database. The operations perform the actions of generalization in support of data reduction in the database. The processes of generalization should be performed a series of operations under the control of constraints. The constraints in database generalization are still lack of research. There is also 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 based on 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 concentrate on in our paper.

Keywords: Constraints, Database generalization, Operations, Spatial Object,Hierarchy.

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 word, this transformation is deemed as changing state of database from one state to another state.

The contents of OO-database mainly consist of objects and relations among objects (implicit or explicit) from a user 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 that 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 word, the transformation of the data schema, object characteristics, relations among objects are the main contents 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).

Constraints in database generalization are still lack of research on database generalization. We can consider constraints as the specification of conditions of data schema, objects and relationships that 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 provide a steering parameters for how many object tyes should apear in a database given a particular context and resolution.2) detect and identify objects violated constraints that have to be generalized. 3) To guide the choice of operators according to constraints priorities; 4) To control the effect of an algorithm.

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 features types (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).

The paper is organized as follows: Section 2 presents a set of constraints that can control the process of generalization in the context of categorical database. Section 3 gives a set of operations in database generalization. Section 4 designs the representation of constraints and operations. Section 5 describes the framework of constrains and operations in database generalization. Section 6 concludes the paper with direction for future work.

2. Constraints to Spatial Database Generalization

Constraints must be identified which are pertinent to database generalization in the sense that constraints provide steering parameters for how many object types should appear in a database given a particular context and resolution and determine which objects can be retained after generalization in the database. One must also identify connections between constraints. The goal of this section is to identify constraints that apply 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 in rest of this section though the list is not exhaustive.

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

2.1.Constraints to Spatial Data Schema

A spatial data schema is a chart of the types of data that are used. It gives the names of the entities (classes) and attributes, and specifies the relations between them. A view is a part of the database for a specific application. It is a schema with an instance set. The spatial data schema 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. It determines the generalized result and the contents of the database. Lanza an La Barbera (1993) proved, in this respect, that a water network can be generalized differently, according to the criteria of hierarchy model used. Data model comprises a data scheme describing what should be included in the model. A schema is a specification of a data view expressed in the language used by a particular database management system. The data model is higher-level descriptions of the data than the schema does. The data model can be mapped into the schema using the languages provided by a particular DBMS.

In the relational model, a schema is the set of attributes for a relation, defining the format of the data, while in Object oriented data model the objects have common patterns of both state and behavior within the framework of an application, they may be grouped into classes to form object types, and object types in turn may be organized into superclasses to form super-types, and so on. So the schema should describe both the structure and behavioral aspects of the data. The schema is mainly a class hierarchy in which each class includes attributes and methods with their implementation.

In the categorical database, the spatial data schema is a kind of multi-level (hierarchical) schema. It could include classification hierarchy and aggregation hierarchy. Both hierarchies reflect a certain aspect of data abstraction. They 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 and Smaalen 1996). Object types, together with classification and aggregation hierarchies are important aspects in semantic data modeling and play a critical role in defining the constraints in database generalization, in a sense that the constraints to classification hierarchy or aggregation hierarchy are also the ones to the schema in a database. In other word, we can discuss the constraints to the schema through the ones to classification hierarchy and aggregation hierarchy.

2.1.1.Classification hierarchy constraints

Object types and super-types can then be organized into a hierarchical structure called classification hierarchy (Smith and 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 deriving aggregation hierarchy and constraints. The top-down relationship of classification hierarchy is called IS-A links. This relationship makes it possible to transform the more complex model to the 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 exiting database according to application requirement in order to generalization, 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 in some application. Because object types at different levels in a classification hierarchy correspond to data of different complexity. In this sense, specifying an (elementary) object type implies, to a certain extent, 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 which control the processes and abstraction levels of generalization may include:

- Hierarchical structure associated with existing database;
- Lower levels in the hierarchy 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;
- Level in which an object type is located in its associated classification hierarchy corresponding to the degree of abstraction;
- 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;
- Number of elementary object types;
- Number of attributes contained in an object type;
- Attribute structure of each object type;
- Similarity evaluation among objects, object types, sub-object types, and super object types;
- One object only belonging to a class and a super class, and one class having many objects;
- Specifying a specific object type and its constituent parts at different levels.

3.1.2. 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 part of the composite-type is called component-type. 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. An 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 a specific composite object and on to a specific 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:

- Hierarchical structure associated with existing database;
- Composite-types in the hierarchy corresponding to higher abstraction levels and resulting in less complex data;
- Component-types correspond to lower abstraction levels and resulting in more complex data;
- Level in which an object type is located in its associated aggregation hierarchy corresponding to the degree of abstraction;
- 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;
- Number of attributes contained in an object type;
- Number of composite-type;
- Number of component-type;
- Similarity among component-object ,composite-object, component object type and composite objects;
- Specializations (rules) specifying the component types of the component objects building an composite object of this type;
- Specializations (rules) specifying the geometric and topologic relationships among these objects;
- Specifying a specific composite object and its constituent parts at different levels.

2.2. Constraints to Objects

The spatial object is an instance of an object type. Object types are classes of spatial entities in a geo-spatial model. In realty, they may be a road, city, river, land use and so on. Three types of constraints can be identified based on the characteristics of spatial objects. They are thematic constraints, geometric (spatial) constraints and temporal constraints. Temporal constraints and related aspects are not discussed in this study.

2.2.1.Thematic Constraints

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

- The same geo-phenomena should be described using the same thematic resolution through out the entire database;
- No object has common boundaries with other objects having the same object type. If case occurs, the separating boundary is dropped;
- Connected (adjacent) objects belong to the different object type may be aggregated;
- Adjacent (disconnected or connected) objects having the same object type or different object types may be merged;
- adjacent (disconnected or connected) objects having the different object types may be aggregated;
- The area of eliminated object should be added to the area of the object which has 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 same similarity with it;
- 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, order, interval and ratio, but cannot 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 word, they are application-dependent. It comprises:

- Geometric description type;
- Minimum object size (minimum size for area objects, or minimum length for line objects);
- Minimum object’s detail that a database can contain
- Objects which are too narrow;
- Minimum space between objects;
- Preserve the global shape and angularity of objects;
- Preserve typical shapes and angularity of objects of each object types;
- Preserve the given size distribution of objects for each objects types;

These two aspects of constraints of spatial objects apply 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. They are spatial relations and semantic relations between objects. Spatial relations are classified into topological relations, direction relations and distance relations. 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 Formal Data Structure (FDS) for single valued vector map, we use topological relationships (connectivity, adjacent, inclusion), distance and directional 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 (1991), spatial relations between objects present constraints for each generalization operation. Spatial and semantic relations constraint govern the process of database transformation.

2.3.1. Topological Relations Constraints

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

- Topological constraint*s* deal with basic topological relationships like connectivity, adjacency and containment, which should be maintained when generalizing data.
- Self-intersection and overlapping objects do not exist and cannot be introduced with generalization.
- An object must not move across the boundary of another object;
- An object must not overlap with another object;
- Avoid introduction of illogical neighborhood relations (e.g., house in a lake);
- Avoid separation of object when deleting parts of it;
- Avoid introduction of self-intersection of object outlines;

2.3.2. Directional Relation Constraints

Cardinal directions describe, qualitatively, the orientation between objects (Frank 1991). The directional relationship between two objects is an important spatial property and can also be used as selection criterion for retrieving objects from a database. A set of spatial objects distributed having certain alignments and patterns along a certain direction in a database reflect spatial distribution property of some geographical entities. These alignments and patterns of the objects should be preserved after generalization. The direction relation constraints may include:

- Preserve typical alignments and patterns of objects within a group of objects in space;
- Preserve the relative location of one object in relation to other ones in direction. After generalization;
- Objects of the same object type may be amalgamated if they are distributed along a certain direction and the distance between them is less than the minimum space;
- Objects of the different object type may be aggregated if they are distributed along a certain direction and the distance between them is less than the minimum space.

2.3.3. Distance Relation Constraints

Distance relation can be described as the approximation among objects in the database.
The distance constraint specifies the distance between two objects that can be merged in database generalization. Distance relation constraints may contain:

- Avoid merge two adjacent objects of the same type which the distance between them is larger than the minimum space;

- Objects of the different object type may be aggregated if the distance between them is less than the minimum space;
- Objects of the same object type may be amalgamated if the distance between them is less than the minimum space;
- Objects of the same object type or the different object type may be merged if the distance between them is less than the minimum space.

2.3.4. Semantic Relation constraints

Semantic relations are also of essential importance to reduce the number of objects in a 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 to indicators. It contains:

- Objects with a sub object type having IS-A relationship may be amalgamated to form the object of the higher object type;
- Objects with a component object type having PART-OF relationship may be aggregated to the composite object of the composite object type;
- Objects with a composite object type having PART-OF relationship may be aggregated to the composite object of the higher composite object type;
- A set of small objects having the same similarity in semantic aspects may be merged to a larger object;
- An object having a specific function should be maintained.

3. Operations in database generalization

A spatial operation is an action changing the state of the representation of a real world object being modeled, or deriving additional information from the current representation. Reducing the number of objects in database generalization is main task (Molenaar 1998, Weibel 19). 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. A reduction in the number of objects can also be accomplished by replacing several objects with one new object. We can define six operations in database transformation based on the geometric, semantic relations and constraints.

- 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 based on the requirements of application and constraints;
- 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 the attributes may need to be modified after aggregating two objects or more into a larger one.
- Merge: a merge operation that merges two or more connectivity 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 the attributes may need to be modified after merge two objects or more into a larger one.
- Amalgamation (Generalization): a 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 the attributes may

need to be modified after generalization two objects or more into a larger one.

- 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).
- 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 unchanged object type.

4. Representation of constraints and operations

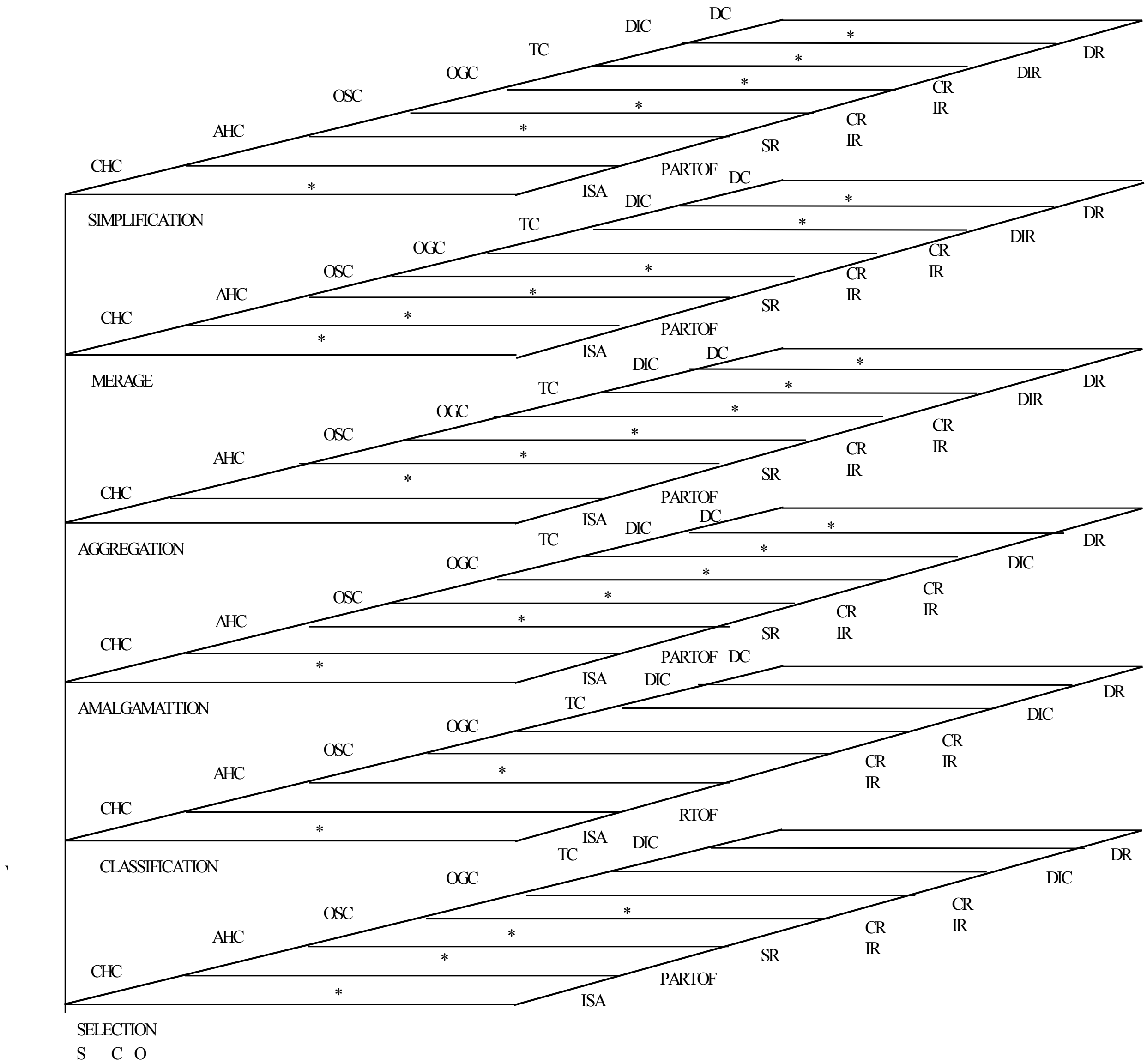
Constraints allow us to indicate where an action should be performed and which operation should be taken to constraint violation. If we want to use constraints violates as triggers, we need to represent them in the database. These constraints apply partly to object, partly to object type level. Some of constraints apply partly to objects of an object type, partly to the entire database, and they may take different values for different object types in the same database.

- Constraints relate to an object can be represented by means of attributes at the object level (e.g. area too small, line too detailed) with either a flag or a quantitative value which describes the severity of the violation;
- Constraints related to a class of objects can be represented at the class level or by means of a specific attribute which is an constraint table that should be consulted during the process (Ruas 1998);
- Operations as the methods of the object can be encapsulated in classes.

5. 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 violates a size constraint, two objects too close violate a distance constraint. An object violating the constraint requires some operation, but several operations are possible to resolve it. 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. Figure 1 shows the diagram among operations, constraints and relations.

Constraints and operations contains three hierarchic levels : the geometric, spatial relation and semantic levels. A classification of these three levels could be from the concrete to the abstract : the geometric level is the inferior one. The intermediate level is the spatial relation level. The semantic level is the superior one. The semantic level contains all the semantic information which are linked to the geometric data. This level is used to define the objects, object type and the attributes of the objects in a database. So the priority order from bottom to top is decreased based on semantic first. Each layer represents one operation. The sign * represents involving the constraints and relations for an operation in a layer.



Legend:

Constraints	Relations
CHC: Classification Hierarchy Constraints	ISA: IS-A
AHC: Aggregation Hierarchy Constraints	PARTOF: PART-OF
OSC: Object Thematic Constraints	SR: Semantic Relation
OGC: Object Geometric Constraints	CR: Connectivity Relation
TC: Topological Constraints	IR: Inclusion Relation
DC: Distance Constraints	DR: Distance Relation
DIC: Directional Constraints	DIC: Directional Relation

6.Conclusion

The constraints as transformation conditions play a key role in the process of database generalization. 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 that database generalization process are very flexible/adaptive, and decision-making is based on 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 database generalization process. The modeling of operations and constraints and relations are identified.

Acknowledge

Research work was supported by International Institute for Aerospace Survey and Earth Science, Ministry of Education PR.China and National Bureau of Surveying and Mapping.

Reference

Beard, K. and Mackaness.W. 1991, Generalization Operations and Supporting Structures, AUTO-CARTO 10, pp.29-42.

Beard, K., 1991, Constraints on Rule Formation, in Map Generalization: Making Rules for Knowledge Representation (eds Buttenfield, B. P. and McMaster, R. B.), Longman House Essex, United Kingdom, pp.121-135.

Beat, P. and Weibel, R., 1999, Integrating Vector and Raster-Based Techniques in Categorical Map Generalization, Third ICA Workshop on Progress in Automated Map Generalization, Ottawa.

Egenhofer, M.J. and Herring, J.R., 1990, A mathematical frame work for the definition of topological relationships. In Brassel, K. and Kishimoto,H. (Eds.), Procedding of 4th International Symposium on Spatial Data Handling, Zurich, PP.803-813.

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

Laurini, R. and Thompson, D., 1993, Fundamentals of Spatial Information Systems, London: Academic Press.

McMaster, R. B., 1991 Conceptual frameworks for geographical knowledge, in Map Generalization: Making Rules for Knowledge Representation (eds Buttenfield, B. P. and McMaster, R. B.), Longman House Essex, United Kingdom, pp. 21-39.

McMaster, R., and Barnett, L., 1993, A Spatial-Object Level Organization of Transformations for Cartographic Generalization, AUTO-CARTO 11,Minnesota,pp.386-393.

Molenaar, M. 1991, Terrain objects, data structures and query spaces, in Geo-Informatik, (ed Schilcher, M.), Siemens-Nixdorf Informationssysteme A.G., Munchen, 1991, pp. 53-70.

Molenaar, M. 1993, Object hierarchies and uncertainty in GIS or why is standardisation so difficult, GeoInformations-Systeme, Vol. 6, No. 3, pp. 22-28.

Molenaar, M. 1996, Multi-scale approaches for geo-data, International Archives of Photogrammetry and Remote Sensing, Vol. XXXI, Part B3, Vienna, Austria, pp. 542-554.

Molenaar, M., 1989, Single valued vector maps- a concept in GIS. GeoInformationssysteme, 2, 18-26.

Molenaar, M.,1996, The Role of Topologic and Hierarchical Spatial Object Models in Database Generalization, In Molenaar,M.(Eds.) Methods for the Generalization of Geo-databases, Delft: Netherlands Geodetic Commission, New Series, Nr 43,pp.13-36.

Molenaar,M. 1998,An Introduction to the Theory of Spatial Object Modelling for GIS, Taylor &

Francis Ltd. London.

Muller, J. C. 1991, Generalization of spatial data bases, in Geographic Information Systems (eds Maguire, D. J., Goodchild, M. F. and Rhind, D. W.), Vol. 1, Longman Scientific & Technical Ltd, New York, pp. 457-475.

Peng, W. 1997, Automated Generalization in GIS, PhD Thesis, Wageningen Agricultural University (and ITC), The Netherlands.

Peuquet, D. 1984, A conceptual framework and comparison of spatial data models, Cartographica, Vol. 21, No. 4, pp.66-113.

Pilouk, M. 1996, Integrated modelling for 3D GIS, PhD Thesis, Wageningen Agricultural University, The Netherlands.

Richardson, D. E. 1993, Automated spatial and thematic generalization using a context transformation model, PhD. Thesis, R&B Publications, Canada.

Ruas, A. and Lagrange, J.P. 1995, Data and knowledge modeling for generalization, in GIS and Generalization (eds Muller J.C., Lagrange, J.P. and Weibel, R.), pp.73-86.

Ruas, A. and Plzanet, C., 1996, Strategies For Automated Generalization, SDH 96, pp.6A1-15.

Ruas, A., OO-Constraint Modeling to Automate Urban Generalization process, SDH 98, pp.225-235.

Shea,K.S., and McMaster,R.B, 1989, Cartographic Generalization in a Digital Environment When and How to Generalize, AUTO-CARTO 9, Baltimore, Maryland,pp56-65.

Smith, J.M. and Smith, D.C.P, 1977, Database abstractions: aggregation and generalization, ACM Transactions on Database System, 2, pp.105-133

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

van Smaalen, J. W. N. 1996, A hierarchic rule model for geographic information abstraction, SDH 96, pp. 4B.31-4B.41.

Ware, M.J., and Jones, Chris.B., 1996, A Spatial Model For Detecting (and Resolving Conflict Caused By Scale Reduction. 7 th International Symposium on Spatial Data Handling, Delft, The Netherlands, pp. 9A15-23.

Weibel, R. 1995, Three essential building blocks for automated generalization, in GIS and Generalization (eds Muller J.C., Lagrange, J.P. and Weibel, R.), pp. 56-69.

Weibel,R. and Dutton, G.H. 1998, Constraints-Based Automated Map Generalization. Proceeding of the 8 th International Symposium on Spatial Data Handling, Vancourver,BC,pp.214-224.

Weibel,R., 1996, A Typology of Constraints to Line Simplification, SDH 96, pp.9A1-13.

Worboys, M.F., 1995, GIS-A Computing Perspective, London: Taylor & Francis.