MODELING MULTIPLE REPRESENTATIONS INTO SPATIAL DATA WAREHOUSES: A UML-BASED APPROACH

Bédard Yvan, Ph.D, Marie-Josée Proulx, M.Sc, Suzie Larrivée B.Sc., Eveline Bernier, M.Sc.

Centre for Research in Geomatics Laval University, Quebec City Canada G1K 7P4 {yvan.bedard, marie-josee.proulx, suzie.larrivee, eveline.bernier}@scg.ulaval.ca

KEY WORDS: GIS, multiple representation, spatial database modelling, spatial data warehouse, spatially-extended UML, Perceptory.

ISPRS Commission IV, Working Group IV/1

ABSTRACT: Creating database schemas that include complex geometries and that take into account map generalization as well as multiple geometric representations is quite a challenge. The created schemas are either too complex to become useful, too large to be usable, too detailed to be checked with the overall integrity in mind, or even too labor-intensive to even be created or edited. This paper presents a method and a tool that has been used with success in modeling generalization and multiple representation for a spatial data warehouse.

RÉSUMÉ: La création de schémas de bases de données qui incluent des géométries complexes et qui tiennent compte des processus de généralisation aussi bien que des représentations géométriques multiples représente tout un défi. Les schémas ainsi créés sont soit trop complexes pour être utiles, trop volumineux pour être utilisables, trop détaillés pour être vérifiables dans une optique d'intégrité générale du schéma, ou même trop exigeants à produire ou éditer. Cet article présente une méthode et un outil qui ont été utilisés avec succès pour la modélisation de la généralisation cartographique et la représentation multiple dans le cadre de création d'un entrepôt de données spatiales.

1. INTRODUCTION

Organizations involved in map production regularly provide several products at different map scales or for different purposes. Ideally, they maintain only one cartographic data warehouse and derive the different products from this warehouse. Since these products meet different goals, they often need to represent the same geographical features in manners that differ in their level of detail, their precision, their symbology, their aggregation and generalization states, etc. The diversity of products stemming from this unique dataset calls for efficient data processing if one wants to derive automatically every product. In fact, in many cases and more specifically when using semi-automated procedures, it becomes more efficient to keep in the database the results of the derivation, leading to several geometric representations for a same feature, i.e. to multiple representations. In some cases, it may even be impossible to derive the desired geometry of an object and additional data acquisition becomes required, also leading to the storing of multiple representations in the database.

From a data modelling point of view, this represents an interesting challenge since the level of complexity can become so high that the resulting database schema rapidly becomes unusable. In order to solve this problem, we propose a solution based on the UML object class model (Unified Modelling Language) that has been extended to better accommodate spatial data. The proposed solution uses UML stereotypes to represent in a very simple manner all the possible 0D, 1D and 2D geometries that an object may have (simple, complex, alternate, multiple). This solution also supports derived geometries as well as the derivation of attributes, object classes or associations. It has been implemented in a free visual modeling tool called Perceptory and the mapping to ISO/TC211 as well as OGC geometric and topological primitives has been done. The complete solution has been applied to the design of the provincial topographic spatial data warehouse of the Quebec Department of Energy and Resources. Thus, the present paper will start with an overview of some fundamental concepts of multiple representations and spatial data warehousing. We will also introduce a spatial database modelling solution we have developed: Perceptory. Then, we will describe the approach we have used with Perceptory to facilitate the modelling of multiple representations and generalization; in particular, we will describe how, using object-oriented concepts and examples extracted from our project with the Quebec Topographic Data Warehouse, we have combined multiple representations with automatic derivation in the database schema. Finally, we will discuss about the presented solution and conclude on this work.

2. FUNDAMENTAL CONCEPTS OF MULTIPLE REPRESENTATIONS AND SPATIAL DATA WAREHOUSING

Multiple representations result from seeing the world from different abstraction levels as well as different points of view. In fact, each user of a spatial database has its own needs regarding the representation of the objects. These representations may vary depending on several factors such as the user's needs or the level of details desired. Storing these multiple representations in order to better fulfill the needs of spatial databases users has been a challenge since many years.

Insofar, research in multiple representations has been focussing mostly on developing data structures. In most cases, these structures allow an object to have different geometries that only vary in their scale. We term them "multiscale structures".

Some of the most recent research projects aim at combining on-the-fly generalization with multiple representation structures [Bernier *et al.*, 2002; Cecconi 2001, Cecconi 2002]. Doing so results in a more flexible solution that can offer products adapted to the user's requirements and at any scale (given the on-the-fly generalization process). These solutions are especially useful for applications such as web-based on-demand mapping and spatial on-line analytical processing (SOLAP).

Of particular interest are the research projects using spatial data warehousing architectures to combine the above-mentioned solutions. A typical solution is a two-tiers data warehousing architecture where a multi-representation data warehouse is fed by independent data production systems and, where possible, by cartographic generalization processes. These acquired/derived data produce multiple representations of a same object (ex. a small town represented by a detailed polygon at 1:20000, a simplified polygon at 1:100000, a demographic centroid at 1:500000) that is then queried to provide only the appropriate geometry for the map being delivered on-line or being printed. In order to facilitate and accelerate the process for the end-users of the warehouse data, three-tiers infrastructures using datamarts can be built around pivot scales (ex. 1:50000, 1:100000, 1:250000) and pivot-themes (ex. transport) that provide "almost-ready" ranges of maps that need only minimal on-the-fly processing to satisfy the users (Bédard et al 2001a). These datamarts are used to quickly derive topic-oriented maps at scales compatible with the pivot scale of the datamart, mostly (but not uniquely) using automatic generalization capabilities (ex. producing a road map at 1:125000 from the transportation 1:100000 datamart). This differs from the data warehouse that relies primarily, but not uniquely, on multiple representations to support the pivot datamarts and to provide object-referencing capabilities across representations. In such three-tiers architecture, automatic derivation is used wherever possible to reduce data updating problems across representations and data acquisition costs; otherwise multiple representations are used.

Finally, few research projects have tried to find a flexible manner to conceptually describe multiple representations and generalization (ex. Proulx et al 2002; Spaccapietra, Parent and Vangenot 2002). Based on a case tool called Perceptory, we have developed a new solution for the modelling of spatial databases that support multiple representations and generalization processes.

3. PERCEPTORY: A SPATIAL DATA MODELLING SOLUTION

Perceptory is a visual modelling tool for designing conceptual object class models based on the UML object-oriented formalism and which includes extensions that facilitate the modelling of spatial databases. Perceptory includes several UML basic components (package, class, attribute, operation, association, multiplicity, aggregation/composition, generalization/ specialization, comments, etc.) as well as formal extension components (called stereotypes) that allow the modelling of spatial characteristics of cartographic objects in a simple manner (Bédard, 1999a) These extensions can be used within the three sections of any instantiated, abstract and associative class, i.e.:

1. at the class name level, to depict the geometry of the objects of a class;
2. at the attribute level, to describe the geometry of the descriptive characteristics within objects of the class; and
3. at the operation level, to describe the initial or the resulting geometry of a process occurring on the geometry of the class.

These stereotypes are fully illustrated by pictogram embedded in a font called PVL (Plug-in for Visual Language). This font can be used by any case tool, including Perceptory. However, the latter uses it in a much more simple and elegant manner.

Perceptory proposes a schema and a data dictionary that includes UML components and some ISO/TC211 (19103, 19107, 19110, 19115) standard components. The UML schema extended with the PVL illustrates the content of the future database while the dictionary describes in more details the schema components with a natural (eg. english or french), formal (eg. Z, OCL) or coded language (eg. datatypes, nulls). The spatial PVL was especially built to visually describe, in a spatial database conceptual schema, the geometric dimension (0D, 1D, 2D) of an object to appear on a map, either following digitizing or derivation. The spatial PVL is also used to visually describe the geometric dimension or spatial extent of an attribute value that is heterogeneous within the boundaries of an object (ex. the attribute "number of lanes" of the class "road" may change within an instance of road and be located with a linear referencing system using "from" and "to" distances from an anchor point).

The three basic geometries of the spatial PVL can be used alone to describe a simple geometry or can be combined to describe an alternate, a complex or a multiple geometry. Each semantic occurrence of a spatial object class can be represented either by:

1. a simple geometry, i.e. composed uniquely by one occurrence of a geometric object;
2. a unique geometric aggregate, i.e. a simple aggregate geometry composed by geometric objects of the same dimension or a complex aggregate geometry composed by geometric objects of different dimensions;
3. an alternate geometry if the geometry can be represented by one dimension or another. The proposed geometries can be simple, simple aggregated or complex;
4. several independent geometries, where usually only one is used at a time. Thus, we say that the geometry of this object class is multiple. These geometries can be simple, simple aggregated, complex or alternatives.

The following figure (figure 1) shows the hierarchies between the different types of geometry used by Perceptory and gives an example for each one.

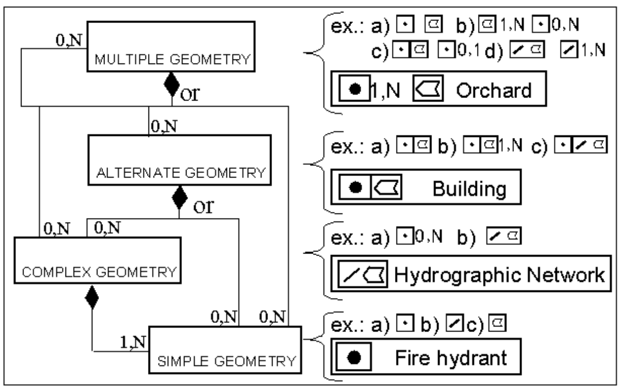

Figure 1. *Examples of possible geometry types (OR is and inclusive OR).*

The geometry can be described in much more details in the dictionary of Perceptory. We precisely define each geometry of an object, i.e. the data acquisition specifications (the "what" and not the "how"), we also include additional information depending on the geometry type (point, line, polygon) as well as the minimal geometric dimensions required for digitalizing the object (ex. point if area < 1000, polygon if area > 1000).

Just like the class attributes, the geometry can also be derived. The pictograms are then in *italic*, which visually remind the slash (/) of UML normally used in front of derived attributes. To do so, the derivation process must be entirely automatic, i.e. without any manual processes. In the case of multiple or alternative geometries, it is possible to have only one derived geometry.

Although an object can have different geometries for the same scale (for example, a house can be mapped according to its roof or its foundations) the kind of multiple representations that we are interested in are the multiscale representations. The following figure is based on a spatial database that can produce paper maps at the scales of 1:1000 and 1:20 000. At the first scale, all the buildings are represented by a polygonal geometry while at the second scale (the smaller one), only the public ones are represented by a derived polygonal geometry. Besides, roads have a polygonal geometry at the scale of 1:1000 and a derived linear geometry at the scale of 1:20 000.

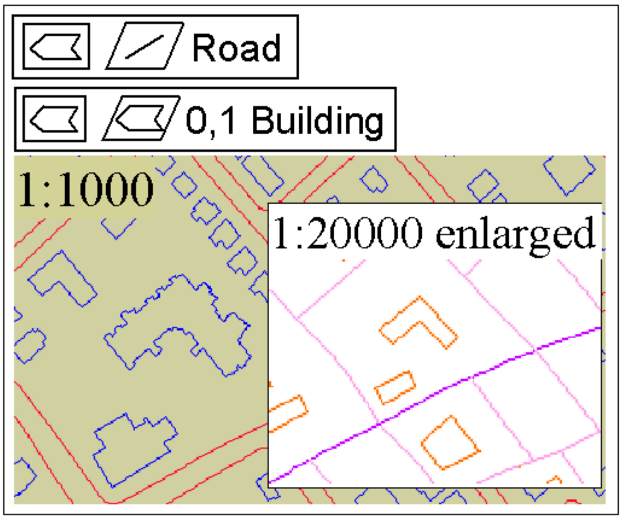

Figure 2. *The modelling of an object class that is represented at two different scales*

4. THE MODELLING OF MULTIPLE REPRESENTATIONS AND GENERALIZATION IN PERCEPTORY

There is a lot of different ways to model cartographic generalization. From a conceptual point of view, this process is expressed by operations on the geometry of an object. The result can be stored (leading to multiple representations) or not. Different possibilities must then be supported in the modelling approach:

- automatic on-the-fly generalization that doesn't result in the storage of multiple representations;

- automatic generalization that leads to stored representations for performance reasons;

- semi-automatic generalization that also leads to storing the results (to avoir repeating the operation);

- manual generalization that leads to storing the results;

- several data acquisitions for the same object in order to obtain the representations needed for every scale.

4.1 General modelling rules

According to our approach, the generalization operations are modelled the same way as any other operation in an object-oriented model. In addition, given the typical rules that govern the modelling process, we only model the data and the operations that must be programmed in the database, i.e. we don't represent the operations used to populate the database. Thus, in theory, we only model on-the-fly generalization (i.e. automatic generalization where the result is not stored in the database) without modelling the generalization operations that leads to multiple representations in the database. Nevertheless, because automatic generalization is very context-sensitive and it is difficult to distinguish in advance what will require human assistance from what can be done automatically (ex.with small programs triggered by the objects), it is possible to include in the first iteration of the model all generalization operations and, after some testing, keep in the last iteration of the model only those operations that can be done fully automatic. By applying these modelling rules, we obtain a database schema corresponding to the implemented data warehouse.

4.2 Specific modelling rules

The modelling rules that express generalization and multiple representations according to our approach are hereunder:

<u>Rule #1</u>: all the generalization operations are defined by an operation in the object class (in the operation section) whether these processes are manual, semi-automatic or entirely automatic (keeping only the automatic ones for the last iteration);

<u>Rule #2:</u> all the multiple representations are defined by the PVL using a multiple geometry pictogram in the object class;

<u>Rule #3 :</u> if the geometry resulting from a generalization process (manual, semi-automatic or automatic) has to be stored in the system, this geometry will be added in the object class with the PVL multiple geometry pictogram.

As previously mentioned, when building the schema, it can be useful to proceed by iteration and to retain only the operations that will be implemented to generalize automatically the object classes of the database.

4.3 The graphic notation

Following is the graphic notation used to model the above rules.

<u>Generalization operations:</u>

- if the generalization process relies on a unique operator, the name of this operator is used as the operation identifier (eg. collapse, simplification, displacement). If the generalization is done by applying a set of generalization operators, we use a general operation called “GEN” and we describe the operators sequence in the data dictionary.

- we indicate, in the operation, the initial and the resulting geometries with PVL pictograms. The pictogram that represents the resulting geometry is in italic if the process can be partly or completely automated (i.e. derivable) (eg. GEN ->). If generalization is entirely manual, then the pictogram uses a regular style (ex. GEN ->).

- we also identify the scale or the abstraction level for each geometry (ex. GEN 20K -> 100K).

Multiple representations

- all the geometries that can be used to represent an object are illustrated by a PVL multiple geometry pictogram in the object class (upper left corner);

- the geometries are ordered from the detailed level to the more general level (from the largest scale to the smallest one);

- the geometries that result from a generalization process and that are stored in the database are defined by a derived multiple geometry pictogram (see rule #3).

4.4 Application examples

We have applied this modelling method for the design of a spatial data warehouse for the Quebec Ministry of Natural Resources (Bédard & *al.* 2001a).

To better explain the different concepts presented so far, we will use two examples from this application. These examples are based on cartographic objects used in different scales: 20 000, 100 000, 250 000, 1 000 000 and 8 000 000 (i.e. 20K, 100K, 250K, 1M et 8M). The first modelling example shows an object class (waterstream sections) for which all geometries (except the most detailed one) can be obtained from generalization, but some of them are stored in the database because their generalization process is quite complex. The second example presents an object class (airport) for which the generalization process give rise to an object replacement in addition to a geometric change.

4.4.1 Example 1: The modelling of multiple representations and generalization for a waterstream section

Let's take a waterstream section that is acquired for the scale of 20k according to certain geometric rules (eg. is a polygon if its area > 2000m^2 and its minimal width is 20m, is a line if its area < 2000m^2 and its minimal length is 150m). So, in the object class, we will have an alternate linear or polygonal geometry ( ) (figure 3). The geometric rules governing the way each geometry is acquired are kept in the data dictionary.

The geometries at the scale of 20K are then generalized to obtain the geometries used at the scale of 100k. Given that a waterstream sections can have two geometries at the scale of 20k, we thus have two different generalization operations. The first one starts with the linear geometry and results also with a linear geometry. The second one starts with the polygonal geometry and results in either a linear or a polygonal geometry, depending on certain rules. The term "GEN" shows that it is a complex operation that includes more than one generalization operators. The complete set of operators as well as their order are indicated in the data dictionary. In the same way, the polygonal geometry at the scale of 20k is also used to obtain the geometry at the scale of 250k, leading also to a linear or a polygonal geometry according to specific rules. Finally, the conceptual object class depicts two geometries acquired alternatively at the scale of 20k and three generalization operations that produce the geometries for the 100k and 250k scales.

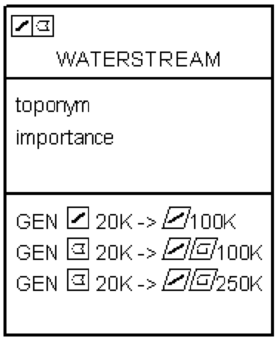

Figure 3. *The modelling of the object class " Waterstream ".*

During the modelling phase, it is possible to find some operations that turned out to be too complex to automate so the resulting geometries should be stored in the database. For instance, let say that it is the case for the generalization process that produces the waterstream section geometry for the scale of 250k. So we just have to add these geometries on the object class (by a derived multiple geometry pictogram, figure 4) to immediately see the multiple representations of this class.

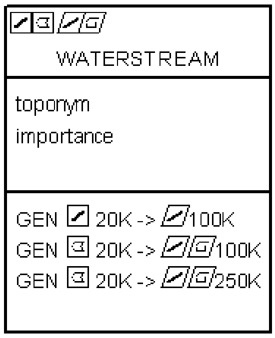

Figure 4. *The waterstream object class after having evaluated the complexity of a given operation*

4.4.2 Example 2: The modelling of a generalization that gives rise to an object replacement

In some cases, an object can have different semantics depending on the scale. For example, at a large scale an airport is represented by its landing strips and its main buildings while at a smaller scale, the same airport is represented by a simple point. At the beginning, there must exist in the database schema a semantic relationship of composition between the landing strips class and the building class with the airport aggregate class. So during the generalization process, the two first objects will be replaced by the third one, i.e. the airport. Given our graphic notation, it is possible to express such a replacement by including the name of the future object directly in the generalization operation of the object class that will be replaced (ex. AIRPORT 250K, figure 5).

First of all, the landing strips are acquired at the scale of 20k. These lines are then generalized for the 100k and replaced by a point airport at 250k. This point airport is then generalized for the scale of 1M.

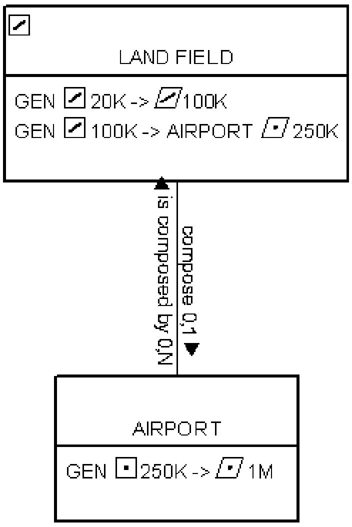

Figure 5. *Example of generalization for airport*

However, in the final database schema, the analyst will probably find more appropriate to store the point geometry of the airport at the scale of 250k instead of systematically derive it. Especially if this geometry is used to produce another geometry at a smaller scale (i.e. the airport geometry for the 1M). Accordingly, the airport in the final schema would have a derived point geometry at the 250k scale (i.e.).

The same example could have been illustrated by point and polygonal buildings that are replaced by an urban zone at a smaller scale.

5. DISCUSSIONS

This new approach has been developed and tested for the modelling of the spatial data warehouse of the Quebec Ministry of Natural Resources (Bédard & *al.* 2001a). Using these simple modelling rules, it was possible to adequately represent the geometries of object classes for the different levels of the data warehouse, that is to say five cartographic scales (20k, 100K, 250K, 1M et 8M). These scales supports, after having been integrated, the objects imported from 10 different products or legacy systems.

Using a traditional modelling approach (based on simple geometric representations and without generalization), we had to model 108 object classes in order to depict the 5 levels of detail of the data warehouse. Obviously, there were several object classes that differed only by their geometries (quarry 20K, quarry 250K, etc.). Although the generalization relations between these classes showed the existence of these types of relation, they lack the richness needed to detail the generalization operations. Also, the schema was difficult to read given the high number of elements (class and relation) when done with a traditional approach (figure 6).

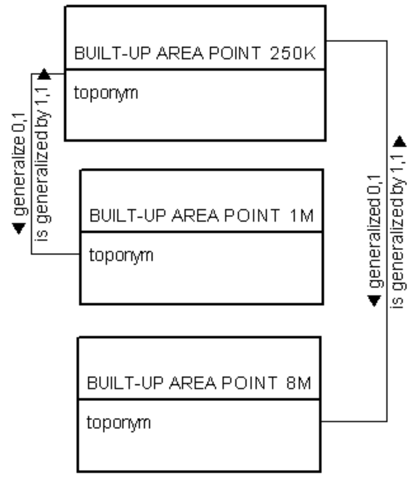

Figure 6. *Example of a traditional approach to model the object class BUILT-UP AREA (250K, 1M et 8M) using relationships.*

Another traditional approach regroups all the different geometries of an object in the form of attributes, which reduces the number of object classes present in the schema. In fact, by using this solution for the modelling of multiple geometric representations and generalization, we reduced the number of object classes in our schema by 40% (i.e. we went from 108 to 66 object classes). This solution is also much more easy to understand since all the representations of a same object are recorded at the same place (i.e. one geographical phenomenon results in only one object class). Furthermore, it would be better to remove the relations between the object classes and describe them as operations in the class (figure 7).

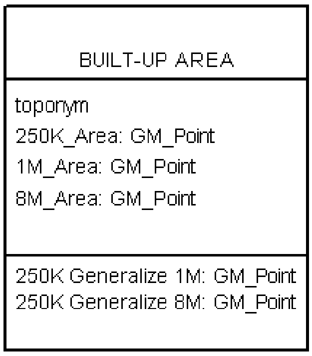

Figure 7. *Example of a traditional approach to model the object class BUILT-UP AREA (250K, 1M et 8M) using attributes and operations.*

Nevertheless, without spatial PVL, the model remains heavy and difficult to read, especially for those who are not familiar with ISO or OCG primitives. The use of PVL pictograms facilitates the reading of the schema while increasing its power of expression. It reduces its size significantly, especially when dealing with alternate geometries. This results in less work to draw the schema, less work to edit the schema, easier reading, easier checking and easier discussions with the clients (fig. 8).

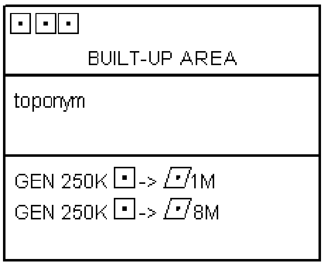

Figure 8. *Example of the used approach to model the object class BUILT-UP AREA (250K, 1M et 8M) with PVL pictograms in Perceptory.*

6. CONCLUSION

We showed in this paper that it is possible to use the spatial PVL in Perceptory to efficiently model geometric multiple representations and cartographic generalization. Given its richness of expression, Perceptory helps analysts in the modeling of multiscale databases. Although we present just a subset of our solution (i.e. geometric multiple representations and cartographic generalization), our approach supports also semantic and temporal multiple representations and generalization. Other works are in progress in related application domains and will help us to improve our solution and make it more flexible and powerful.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the financial support of the GEOIDE Network of Centers of Excellence via the DEC#9 project: *Development of automated techniques to extract, generalize, and access geospatial information from hyperspatial remotely sensed data*. The authors also acknowledge the financial support of the Natural Sciences and Engineering Research Council and of the Quebec Ministry of Natural Resources (DGIG).

REFERENCES

Bédard, Y., 1999a «Visual Modeling of Spatial Databases Towards Spatial Extensions and UML», Geomatica, Vol 53, No2, pp. 169-186, 1999.

Bédard, Y., Proulx, M.-J. Larrivée, S., Nadeau, M., Frenette, P., Rivest S., 2001a « Mise en place d'une infrastructure d'accès aux données de la DGIG- Modélisation des données de l'entrepôt, Avis sur les orientations d'architecture technologique pouvant influencer la modélisation. » Version préliminaire du 21 mai, Rapport de recherche produit pour le Ministère des ressources naturelles du Québec, Centre de recherche en géomatique, Université Laval, 65 pages.

Bernier, E., Bédard, Y, Cecconi, A., Weibel, R., 2002, « La généralisation à la volée », Book chapter submitted to Hermès Edition, Paris.

Cecconi A., Weibel R. «Map Generalization for On-demand Mapping», GIM International, 15(12), p. 12-15, 2001.

Cecconi A., Weibel R., Barrault M. «Improving Automated Generalization for On-demand Web Mapping by Multiscale Databases», 10th International Symposium on Spatial Data

International Organization for Standardization, ISO/TC 211 19103, 2001a « Geographic Information- Conceptual schema language », Draft for International Standard, 75 pages

International Organization for Standardization ISO/TC 211 19107, 2001b « Geographic Information- Spatial Schema », Draft for International Standard, 185 pages

International Organization for Standardization, ISO/TC 211 19110, 2001c « Geographic Information- Methodology for feature cataloguing », Draft for International Standard, 60 p.

International Organization for Standardization, ISO/TC 211 19115, 2001d « Geographic Information- Metadata », Draft for International Standard, 151 pages

Martel, C, 1997. Analyse des capacités et limites de MGE/Map Generalizer pour la généralisation des cartes routières. Département des Sciences géomatiques, mémoire de fin d'étude.

Proulx, M.-J., S. Larrivée and Y. Bédard 2002. Représentation multiple et généralisation avec UML et l'outil Perceptory. Chapitre de livre sur la généralisation cartographique, soumis aux Éditions Hermes, Paris.

Poole, J., Chang, D., Tolbert, D., Mellor, D., 2002. Common Warehouse Metamodel, An Introduction to the Standard for Data Warehouse Integration. Willey & son, 269 pages.

Spaccapietra S., Parent C., Vangenot C. « GIS Databases: From Multiscale to MultiRepresentation», Abstraction, Reformulation, and Approximation, B.Y.Choueiry and T.Walsh (Eds.), LNAI 1864, Springer, 200 (Proceedings 4th InternationalSymposium, SARA-2000, Horseshoe Bay, Texas, USA, July 26-29, 2000).