GIS CONSISTENCY BY ERROR DETECTION AND CORRECTION

Thierry Ubeda
Laboratory for Information Systems Engineering(LISI).
Institut National des Sciences Appliquées de Lyon (INSA).
Bât 404, 20 avenue Albert Einstein
69621 Villeurbanne Cedex - FRANCE.
E-mail : ubeda@if.insa-lyon.fr

ISPRS Commission VI, Working Group 3

KEY WORDS: Consistency Checking, Error Correcting, GIS, Integrity Constraints, Topological Relations.

ABSTRACT:

Data consistency is a main issue of database management systems. Managing geographic data sets brings specific consistency problems related to the spatial attributes of data. Regardless of the data source (map digitization, aerial photos, GPS, etc.), resulting geographic data sets must be consistent in order to be used in spatial analysis. This paper presents a framework to check spatial consistency and to correct spatial errors of GIS data sets in vector format. This framework relies on object's geometry (the shape) and on topological relations between objects. First, a shape admissibility process is applied. It allows to detect inconsistency in shape representation of objects (for example an unclosed polygon). Secondly, topological integrity constraints are defined and applied. They allow to find inconsistent topological scene in the data sets (for example a road inside a lake). The shape admissibility process relies only on the shape of each object without paying any attention to the semantics attached to it. If a lake and a building are both represented by a polygon, they are checked the same way. In contrast, topological integrity constraints rely both on the shape and on the semantics of objects. In the previous example, different constraints will be defined for lake and for buildings. Each inconsistency detected is an error to be corrected. The error correcting process computes a set of correcting scenarios for each error. A scenario is computed by applying a elementary transformation to an object. Each correction proposed ensures not to create a new inconsistency. All the processes presented here have been designed to be suitable for an end-user.

RÉSUMÉ:

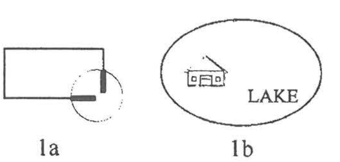
La cohérence des données est un problème majeur des systèmes de gestion de bases de données. La gestion des données géographiques ajoutent des problèmes de cohérence spécifiques liés aux attributs spatiaux des données. Indépendamment de la source utilisée (digitalisation de cartes, photos aériennes, GPS, etc.), les données acquises doivent être cohérentes afin d'être utilisables dans des raisonnements spatiaux. Cet article présente un modèle pour vérifier la cohérence spatiale et corriger les erreurs des ensembles de données géographiques au format vecteur. Ce modèle s'appuye sur la géométrie des objets (la forme) et sur les relations topologiques entre les objets. Dans un premier temps, un processus de vérification de la forme est appliqué. Il permet la détection des incohérences dans la représentation des objets (par exemple, un polygone non fermé). Dans un second temps, des contraintes d'intégrité topologiques sont définies et appliquées. Elles permettent la détection des scènes topologiques incohérentes (par exemple une route dans un lac). Le processus de vérification de la forme dépend uniquement de la forme de l'objet, sans se soucier de la sémantique qui lui est attachée. Si un lac et un bâtiment sont représentés par un polygones, ils sont traités de la même façon. En revanche, les contraintes d'intégrité topologiques reposent à la fois sur la forme et sur la sémantique des objets. Dans l'exemple précédent, des contraintes différentes seront définies pour un lac et pour un bâtiment. Chaque incohérence détectée est considérée comme une erreur qui doit être corrigée. Le processus de correction d'erreurs calcule un ensemble de scénarii de correction pour chaque erreur. Un scénario est calculé en appliquant une transformation élémentaire à un objet. Chaque correction proposée assure qu'elle ne créera pas une nouvelle incohérence. Tous les processus présentés ont été définis dans le but d'être manipulables par un utilisateur final.

1. INTRODUCTION

Data consistency is a prerequisite of spatial analysis. Geographical Information Systems (GIS) stored a great number of data that often contain errors, and especially spatial errors (bad shape, erroneous topological relations, etc.). Those errors make answers to spatial queries or reasoning not reliable and hinder spatial analysis. This is a matter of data quality control.
Nowadays, spatial data are acquired using a lot of different sources: aerial photos, map digitization, GPS, etc. Data are then processed in order to be stored using two main models: vector and raster. For both kinds of model, data sets need to be certified before any reasoning on data.

The goal of this paper is to provide a framework for data spatial consistency checking and error correcting of data sets in vector format. Such a process is to be applied after data acquisition and before spatial analysis. It can also be used to check existing data sets.

Consistency checking will allow to detect errors in the data set. Two kinds of errors are identified: geometric errors (see figure 1a) and semantic errors (see figure 1b).
Geometric errors are based on the shape of



1a 1b

Figure 1 Examples of errors.

geographical objets (point, line, polygon) and the way there are represented in the data model. Those errors will be detected by defining a set of properties on each kind of objet of the data set. Semantic errors come from topological relation between objets (overlapping, inclusion, etc.) and will be detected by defining topological integrity constraints on geographical objets. Those constraints will rely on the shape and on the semantics of objects. The shape will be used to describe the topological relation and the semantics will be used to decide whether the relation is erroneous or not. Because each data set is different from another one, topological constraints cannot be general. This means that they must be defined for each data set by someone who knows well the database (an end-user). For this reason, an easy-to-use interface for topological constraints definition has been designed.

Error correcting is the final goal of this work. This paper proposes a method based on scenarios computation. For each error, a set of possible corrections will be compute. A scenario is the association between a given error and a given correction. All the computed scenarios will be presented to the user that will have to choose one of them.

The paper is organized as follows: in the first two parts, geometric errors and semantic errors will be defined and presented. In a third part, data sets correcting processes will be described. Then we will conclude.

2. GEOMETRIC ERRORS

Previously to the presentation of the geometric errors checking process, a definition of an entity is given. An entity has three Components: a shape (point, line, polygon), semantics (textual attributes) and spatial relations with other objects (based on direction, on distance and on topology).

Geometric errors are errors occurring in the shape representation of an entity. The point is to ensure the spatial representation of the object is consistent regarding to the data model. The process to check the spatial consistency is called the shape admissibility process. It has two parts: the definition of geometric properties and the verification of those properties.

2.1 The definition of the properties

Because the issue is to ensure the shape consistency regarding to the data model, the list of properties to check will be different from a data model to another one. It is then impossible to give a list of properties suitable for all GIS. For this reason, it has been chosen to attach properties to each kind of objects commonly used in vector format (see Figure 3), so one can build the appropriate list of properties for his data model by picking up the relevant properties from the general framework given on Figure 3. This Figure gives commonly used geometric objects and the properties attached to it. Properties can be attached to an object or to a link between two objects.

Depending on his data model, the user will have to choose which properties to apply. In (Plümer 1996), Plümer described a vector data model (planar graph) and gave a set of properties he proved to be complete for his model. His model relies on points, nodes, edges and faces. All the properties he gave can be found on the Figure 3, attached to the objects or to the link between the objects.

The list of properties given on Figure 3 is not proved to be complete for all kinds of vector data model, but can be used for the most common and used ones. Planar graph is one of this. A way to set a complete list of properties will be to defined a classification of vector data model. Frank in (Frank 1983) defined such a classification for model using only points and arcs. Area objects have to be added to the classification for a complete coverage. Unfortunately it has not been done yet.

A study of common data model and the way they can handle those properties can be found in (Ubeda and Servigne 1996a and 1996b)

3. SEMANTIC ERRORS

Semantics errors are defined using the meaning of geographical objects, that is to say the real world entities described by the object. Topological errors are a kind of semantic errors. Topological relations are based only on the shape of objects, but semantics of objects have to be taken into account to decide whether a topological scene is consistent or not. Topological relations are of great importance in GIS (Cui & al 1993). A lot of errors contained in GIS came from erroneous topological relations among geographical objects (Laurini & Milleret-Raffort 1994). Most of GIS do not deal with topological relations, or consider only few relations such as adjacency and inclusion.

In this part, a mean to define topological integrity constraints is described. It allows someone to quickly design constraints for his own data set.

3.1 Topological Model

Topological integrity constraints rely on topological relation. Such relations describe the relative position of objects in the embedding space. We introduce here a model to handle those relations.

This topological model has been designed by Max J. Egenhofer in (Egenhofer & Herring 1990, Egenhofer 1991). In this model, binary topological relations between two objects A and B are defined in terms of the nine intersections of A's boundary (∂A), A's interior (A°) and A's exterior (A^-) with the boundary (∂B), interior (B°) and exterior (B^-) of B (see Figure 2). Each object A and B can be a point, a line or a polygon. Definition of each part of each kind of geometric object is the following:

P is a point :	$P = \partial P = P^\circ$.
L is a line :	∂L = the two ending points of L.
	$L^\circ = L - \partial L$.
Po is a polygon :	∂Po = the intersection of the closure of Po and the closure of the exterior of Po.
	Po° = the union of all open sets in Po.

For each intersection, the value empty (ϕ) or non-empty ($\neg\phi$) is computed and stored into a 9x9 matrix:

$$\begin{pmatrix} \partial A \cap \partial B & \partial A \cap B^\circ & \partial A \cap B^- \\ A^\circ \cap \partial B & A^\circ \cap B^\circ & A^\circ \cap B^- \\ A^- \cap \partial B & A^- \cap B^\circ & A^- \cap B^- \end{pmatrix}$$

ϕ if the intersection is empty
$\neg\phi$ if the intersection is non-empty

Figure 2. The 9-intersection matrix.

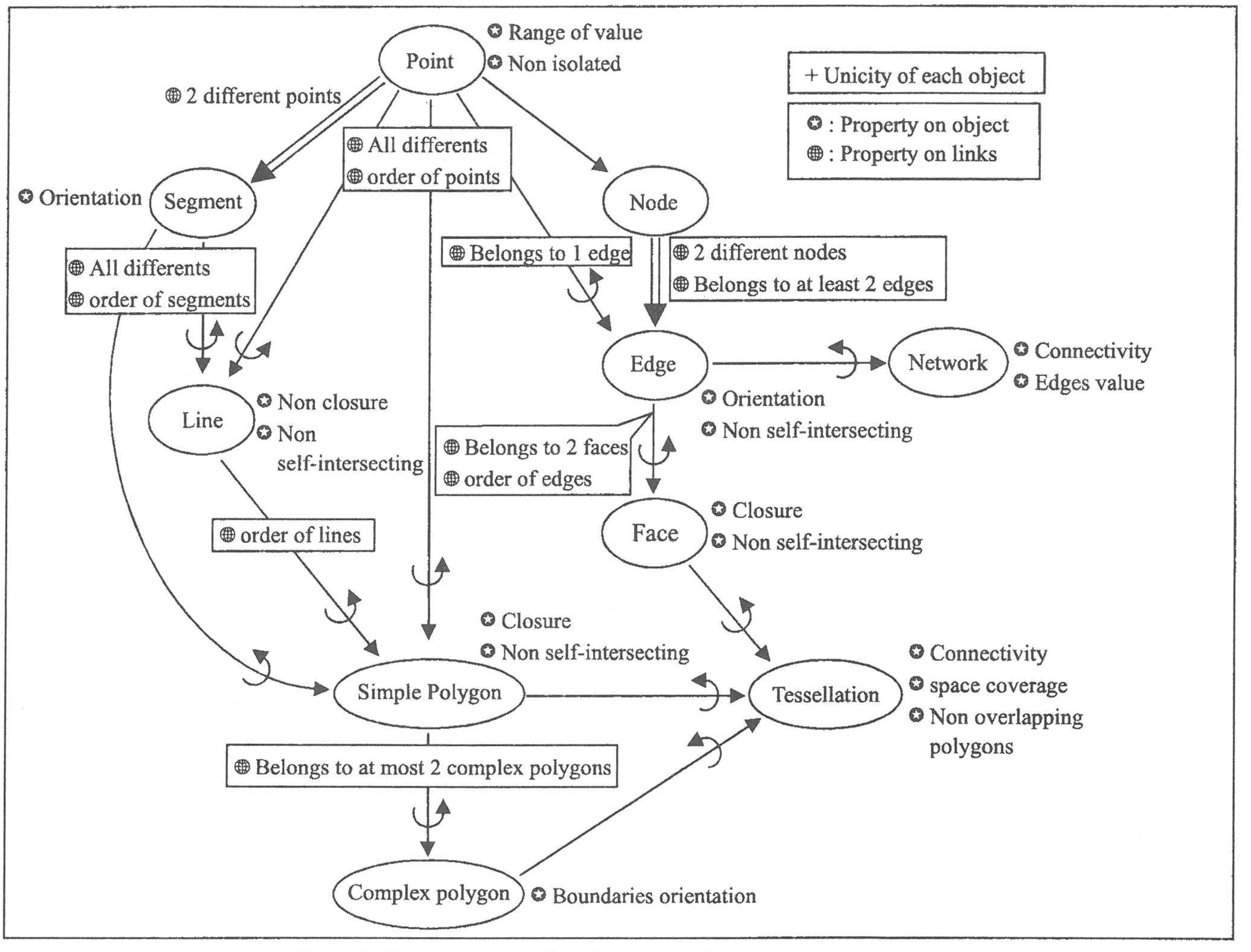


Figure 3. Common geometric objects properties

3.2 Topological Integrity Constraints

Topological integrity constraints are defined using topological relations described by the 9-intersection model. The topological relation between two objects is the main part of the constraint. Considering the shape of objects, it is possible to compute all possible topological relations between two objects (according to the 9-intersection model). Considering the semantics of object (their meaning), it is possible to define which topological relation is consistent and which one is inconsistent.

A topological constraint is defined as the association of two geographical objects, a topological relation between them and a specification (see Figure 4) which can be one of the following:

1. Forbidden
2. At least *n* times
3. At most *n* times
4. Exactly *n* times

Co(Entity class1, Relation, Entity class2, Specification).

Figure 4. The definition of a topological constraint.

The specification *forbidden* is the most interesting and usable one. Topological integrity constraints defined using this specification are a mean for end-users to describe topological situations they do not want to see in their database.

3.3 Groups and subsets of relation

The 9-intersection model can be applied to all kinds of geometric objects. Considering points, lines and polygons, it leads to six groups of relations: point/point, point/line, point/polygon, line/line, line/polygon, polygon/polygon.

In (Egenhofer & Herring 1991), the authors gave the list of relations that can be realized in each group, if objects are embedded in 2-D (see Table 1).

The results given in Table 1 consider two converse relations as only one since it is possible to change A in B and B in A. Converse relations can only happen between two objects of the same kind, namely in point/point, line/line and polygon/polygon groups.

Group of relations	Number of relations
point/point	2
point/line	3
point/polygon	3
line/line	23
line/polygon	19
polygon/polygon	6

Table 1 Number of relations per group

Each one of those relations can be used to define a constraint. The number of possible relations (81) is an impediment to the design of constraints for two reasons: there is too many relations to give a name to each of them, and to avoid a single situation to happen, one will have to create several constraints.

In order to solve both problems, relations of each group have been grouped according to the elements of the 9-intersection matrix (see Table 2). A classification of 2 to 6 subsets per group is then defined. Using a subset in the definition of constraint will make the model more practicable for the user.

3.4 Visual Interface for the Definition of Topological Integrity Constraints

In this section, we present a visual interface to define topological integrity constraints. Specifically, a dialogbox in which the user can choose a pair of entities, a topological relation or a set of topological relations, and a specification (see Figure 5). Topological constraints are defined following the lists of operations given here:

1. Choose a first class of entities.
2. Choose a second class of entities.
3. Choose a relation or a set of relations among the list proposed.
4. Define the specification.

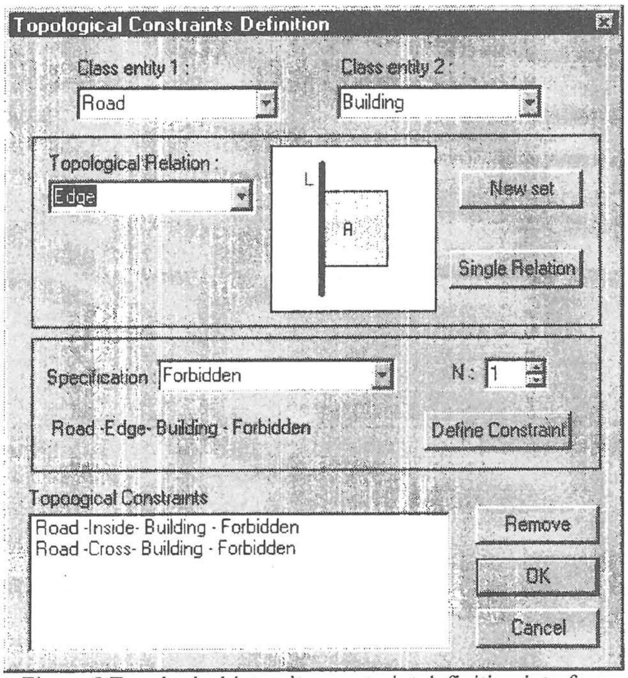


Figure 5 Topological integrity constraint definition interface

In the case shown on the Figure 5, the constraint defined is:
`(Road, Inside, Building, Forbidden)`
where Inside is a set of 10 relations describe by $(A^\circ \cap B^\circ=1)\wedge(A^\circ \cap B^-=\phi)$ (see table 2).

The dialogbox shows a schema that illustrates the topological relation chosen in the constraint definition.
This interface has been designed using VisualC++.

Examples of topological constraints

```
C1(Road, Cross, Building, Forbidden)
C2(Sluice, Joint, Waterpipe, Exactly 2 times)
```

Group object A	Group object B	Subset	Subset attributes
point	point	Equality	$A=B$
		Disjoint	$A\cap B=\phi$
point	line	End point	$\partial A\cap\partial B=0$
		On	$\partial A\cap B^\circ=0$
		Disjoint	$(A\cap\partial B=\phi)\wedge(A\cap B^\circ=\phi)$
point	polygon	Inside	$A^\circ\cap B^\circ=0$
		On	$A^\circ\cap\partial B=0$
		Disjoint	$(A\cap\partial B=\phi)\wedge(A\cap B^\circ=\phi)$
line	line	Cross	$A^\circ\cap B^\circ=0$
		Joint	$\partial A\cap\partial B=0$
		Meet	$(\partial A\cap B^\circ=0)\vee(A^\circ\cap\partial B=0)$
		Covers	$A^\circ\cap B^\circ=1$
		Disjoint	$(\partial A\cap\partial B=\phi)\wedge(\partial A\cap B^\circ=\phi)\wedge(A^\circ\cap\partial B=\phi)\wedge(A^\circ\cap B^\circ=\phi)$
line	polygon	Inside	$(A^\circ\cap B^\circ=1)\wedge(A^\circ\cap B^-=\phi)$
		Outside	$A\cap B^\circ=\phi$
		Cross	$(A^\circ\cap B^\circ=1)\wedge(A^\circ\cap B^-=1)$
		Meet	$(\partial A\cap\partial B=0)\wedge(A^\circ\cap\partial B\neq 1)$
		Edge	$A^\circ\cap\partial B=1$
		Disjoint	$(\partial A\cap\partial B=\phi)\wedge(\partial A\cap B^\circ=\phi)\wedge(A^\circ\cap B^\circ=\phi)$
polygon	polygon	Inside	$(A^\circ\cap B^\circ=2)\wedge(A^\circ\cap\partial B=\phi)$
		Overlap	$(A^\circ\cap B^\circ=2)\wedge(\partial A\cap B^\circ=1)\wedge(A^\circ\cap\partial B=1)$
		Meet	$(\partial A\cap\partial B=0)\wedge(A^\circ\cap B^\circ=\phi)$
		Edge	$(\partial A\cap\partial B=1)\wedge(A^\circ\cap B^\circ=\phi)$
		Disjoint	$A\cap B=\phi$

Table 2 Definition of subsets of relations

4. GEOGRAPHIC DATA SETS CORRECTION

Once all the properties and the topological integrity constraints of the data set have been specified, it is time to check for errors according to those rules.

4.1 Geometric Errors

Each time an object does not respect a property, an error is detected. Data checking and correcting depends on the data model. For most of properties a dedicated algorithm usable in most of data model can be defined (for example to ensure the closure of polygon, it is enough to have three algorithms depending on how the boundary is defined, using points, segments or lines.). Those verifications are a matter of computational geometry. A lot of useful algorithms can be found in (Preparata and Shamos 1986) and (Laurini and Thompson 1992). Without a classification of vector data models in GIS it is impossible to set a complete set of correcting algorithms, nevertheless a lot of simple cases can be handle the same way with well-known algorithms.

Several attempts to design methods to handle geometric properties can be found. In (Plümer 1996) the author presents a set of properties for planar graphs and set of transactions that keep those properties verified. In (Tanzi and Ubeda 1995), algorithms for properties checking and correcting in network data model were presented.

4.2 Semantic Errors

In this section, a method to check and to correct errors coming from violated constraint is presented. **This model deals only with constraints defined using the `forbidden` specification.**

Data checking is the first step. A topological relation between two objects depends on the relative position of their shapes. To simplify the problem, points, lines and polygons (or region) are the only shapes considered in the following.

To verify topological integrity constraints we need to calculate the 9-intersection matrix, or at least some element of it, that is to say the intersections of the boundaries, the interiors and the exteriors of the two objects. Each constraint can then be translated into a conjunction of verifications according to the relation involved in it (see Table 2 for constraints using. subsets, and see Ubeda 1996c for more details on single relations based constraints).

Each subset gave in table 2, or each single topological relation can be described by a partial relation. For example, a line crossing a line is defined by a single element of the 9-intersection matrix: $A^\circ \cap B^\circ = 0$, and leads to only one verification. Each verification is described by a predicate based on the dimension of the intersection. In the previous example: `INTERSECTION_DIM0(A°,B°)`. In order to compute those predicates, functions to retrieved the boundary, the interior and the exterior of objects are required. The complete sentence for the previous example becomes:

`INTERSECTION_DIM0(interior(A),interior(B)).`

Type of predicate	Object 1	Object 2	Predicate name
INTERSECTION_DIM2	region	region	RR_SHARE_R
INTERSECTION_DIM1	region	line	RL_SHARE_L
	line	line	LL_SHARE_L
INTERSECTION_DIM0	region	point	RP_SHARE_P
	line	line	LL_SHARE_P
	line	point	LP_SHARE_P
	point	point	PP_SHARE_P
INTERSECTION_EMPTY	region	region	RR_DISJOINT
	region	line	RL_DISJOINT
	region	point	RP_DISJOINT
	line	line	LL_DISJOINT
	line	point	LP_DISJOINT
	region	point	PP_DISJOINT

Table 3 Predicates definition

Considering four types of predicates and three kinds of geometric objects. 9 functions and 14 predicates are enough to check all possible topological integrity constraints (see Table 3 for the 14 predicates). The 9 functions are:

boundary(point)	interior(point)	exterior(point)
boundary(line)	interior(line)	exterior(line)
boundary(region)	interior(region)	exterior(region)

For example, the following topological integrity constraint has been defined:

`(River, cross, River, Forbidden).`

The relation is translated into:

```
I1 = interior(River1),
I2 = interior(River2).
TR : INTERSECTION_DIM0(I1,I2).
```

And the constraint become:

`if TR then Inconsistency detected.`

Each time an inconsistency is detected, the scene is stored into a logbook that will be used during the data correcting process.

Remark: Under the condition that we can calculate the boundary, the interior and the exterior of each kind of geometric object of the data set, it is possible to check and correct the topological integrity constraints regardless the data model used.

Data correction: The goal of this step is to define a model to compute corrections to topological integrity constraint violations (topological errors). Since an error is defined as a forbidden topological relation between two objects, the way to correct an error will be to change the topological relation between those objects. A set of correcting scenarios will be computed by applying several kinds of changes to both objects involved in the forbidden topological relation (together or one after each other). The changes proposed are the following :

- Objects modification :
  - Moving the objects.
  - Reshaping the objects.
- Deleting one object.
- Object splitting (creating an new object).

Computing and proposing correcting scenarios have two main advantages. The first one is to facilitate and to accelerate the end-user work. The second one is to control the correcting process so that it can be ensured that the correction does not create a new error.

Moving an object ensure that the surface area of both objects remains unchanged. One of the two objects involved in the forbidden relation is moved according to a main direction until the topological relation changes. Applying this method to both ways of each main direction leads to a first set of correcting scenarios.

Reshaping means moving a part of an object, leaving the other part unchanged. The goal of such a correction is to change the topological relation between the two objects without changing the relations with the other objects of the data set. The adjustments will be made by some force-fitting algorithm that will snap characteristic points of one object onto characteristic points of the other object.

Deleting one object is useful when an object has been digitized twice. Two objects very closed to each other can then be found. Two correction are possible:

- keeping A and removing B
- keeping B and removing A

This leads to two scenarios.

Splitting one object into two new objects allows to keep the planarity of a map. The only condition to check is that the two new topological relations are different from the previous one. Such a correction can be proposed when the forbidden relation is such as one of the two objects shares a part of its interior with the interior or the boundary of the other objet. The correction are:

- to split one of the two objects into several parts.
- To create a new object based on the shared part and removing it from each other object.

Figure 6 presents the correcting scenarios for two polygons sharing a part of their interiors.

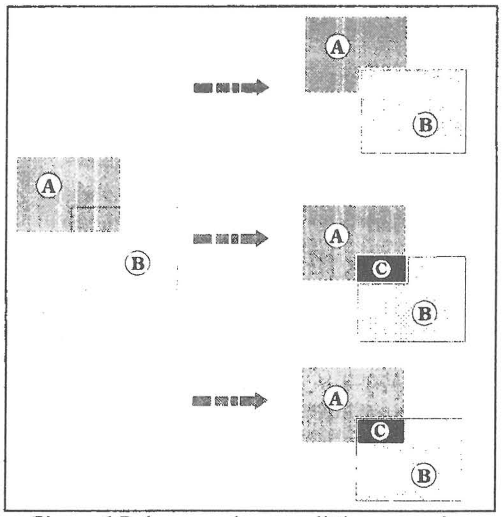


Figure 6 Polygon-polygon splitting scenarios

The first possible correction is to remove the sharing part from one of the object (A of Figure 6). The new topological relation is then Edge. This case covers two scenarios.

The second scenario creates a third object, C, and remove it from both A and B. The new topological relations are:

- Disjoint between A and B.
- Edge between A and C.
- Edge between B and C.

The last one creates a third object named C, and remove it from only one of the two objects (A of Figure 6). The new topological relations are:

- Edge between A and B
- Edge between A and C
- C Inside B.

This case covers two scenarios.

Correcting scenarios presentation: for each topological error, a list of correcting scenarios will be computed. The last step of the correcting process is then to choose which one to apply. To help the end-user to select the appropriate correction, the list of correcting scenarios will be presented using filtering and sorting process.

Filtering process

1. All the correcting scenarios in which the topological relation is used in a topological constraint are removed from the list of corrections. This will be applied when there are more than one constraints defined for a given pair of geographical objects.
2. For correcting scenarios obtained by moving one object, a maximum range is defined. All corrections for which the moving distance is over the threshold are removed from the list of corrections.

Sorting process

The end-user can specify one parameter that will help him to find the appropriate correction. Correcting scenarios in which this parameter is verified are proposed first. The possible values for this parameter are :

1. keeping the area of the object unchanged
2. minimum distance move
3. border adjustment (result of force-fitting first)
4. keeping two objects
5. specifying the new topological relation

Those two process will facilitate the choice of the correcting scenario to apply. If the end-used cannot find an appropriate correction among the list proposed, a set of tools will be provided in order to let the end-user modify manually the geographical objects.

4.3 Handling exception

The goal of geographic data sets is to give a computer representation of the real world. It is based on a data model which contains properties that data must respect. Topological integrity constraints add rules that data must respect as well. Those constraints are attached to a class of geographical objects and force each object of this class (road for example) to respect the same rules. It means that we expect the real world objects to have pre-defined characteristics.

Such an hypothesis is too strong. Exceptions always occurs in geographic data sets. For example, most roads do not cross buildings, but some of them do. It does not mean that we can not use such a constraint, but that we must provide a way to handle exceptions.

The problem of exception to integrity constraints is that situations describing exactly the real world are defined as an error. A solution to this problem is to manage a exception database. Each constraint violation detected by the correcting process can be defined as an exception by the end-user. This takes place when the end-user is asked to choose a correcting scenario. Instead of changing the topological relation, he will have the possibility to add the situation to the database. The same error will then not be reported again.

5. CONCLUSION

In this paper, a complete framework for geographic data sets consistency checking has been presented. It was divided into three steps:

- properties and constraints definition
- properties and constraints checking
- errors correcting

A fourth step that has not been presented here can be added: result certification. The goal of this last step is to ensure that all corrections are consistent. It is straightforward when the correction is made by a scenario, but has to be checked when the correction is made manually by an end-user.

The list of properties given Figure 3 need to be tested with more data models than what have done in (Ubeda and Servigne 1996b). A complete study of data models proposed by GIS software editor (ESRI, Intergraph, etc.) is actually conducted. The goal is to complete the list of object and properties.

The topological clauses presented in part 4.3 are very simple. There are based on only 14 predicates and 9 functions and are not described using a complete mathematical model. The introduction of such a model was useless regarding the goal of this paper. Nevertheless, a complete mathematical formalism based on predicate calculus can be found in (Hadzilacos and Tryfona 1992).

A complete coverage of the correcting scenarios computation processes has been made and will be the subject of future work. This study has been conducted in collaboration with the professor Max J. Egenhofer from the National Center for Geographic Information and Analysis of Maine (NCGIA) and the Department of Surveying Engineering of the University of Maine, Orono.

Parts of this work have been implemented in Visual C++. In the future we will focus on the correction process. This part is currently under development in order to test the correcting algorithms.

Bibliography

Cui Z., Cohn A.G., and Randell D.A., 1993 Qualitative and Topological Relationships in Spatial Databases, *in Proceedings of the Third Symposium on Large Spatial Databases*, Singapore, June 23-25, 1993 (SSD'93). Lecture Notes in Computer Science n°692. pages 296-315.

Egenhofer M. J. and Herring J. R., 1990 A Mathematical Framework for the Definition of Topological Relationships, *in Proceedings of the 4th International Symposium on Spatial Data Handling*, Zurich, 1990 (SDH 90). pages 803-813.

Egenhofer M. J. and Herring J. R., 1991, Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases. Technical Report, Department of Surveying Engineering, University of Maine, Orono.

Egenhofer M. J., 1991 Reasoning about Binary Topological Relations, *in Proceedings of the Second Symposium on Large Spatial Databases*, Zurich, 1991 (SSD'91). Lecture Notes in Computer Science n°525. pages 143-159.

Frank, A. U, 1983. Data Structures for Land Information Systems: Semantic, Topological and Spatial Relations in Data of Geo-Sciences - phD thesis, in German, Swiss Federal Institute of Technology.

Hadzilacos T. and Tryfona N., 1992, A Model for Expressing Topological Integrity Constraints in Georaphic Databases, *in Proceedings of Theories and Methods of Spatio-Temporal Reasonning*, Pisa, september 21-23, 1992 (GIS 92), pages 253-368.

Laurini R. and Thompson D., 1992, Fundamental of Spatial Information Systems, Academic press, 680 pages.

Laurini R. and Milleret-Raffort F., 1994, Topological Reorganization of Inconsistent Geographic Databases : a Step Toward their Certification, *in Computer and Graphics*, Volume 18, number 6, nov-dec 94, p. 803-813.

Plümer L., 1996, Achieving Integrity and Topology in Geographical Information Systems, In: *First International Conference on Geographic Information Systems in Urban Regional and Environmental Planing*,

Preparata Franco P. and Shamos Michael I.., 1986 Computational Geometry: An Introduction. New-York: Springer-Verlag.

Tanzi T. J., and Ubeda T., 1995, Contrôle toplogique de la cohérence dans les bases de données géographiques : application aux réseaux, In: *Revue Internationale de Géomatique*, volume 5, number 2, pages 131-155.

Ubeda T. and Servigne S., 1996a, Capturing Spatial Object Characteristics for Correcting and Reasoning, In: *Proceedings of Joint European Conference and Exhibition on Geographical Information*, Barcelona, Spain, March 1996 (JEC-GI 96), pages 24-33.

Ubeda T. and Servigne S., 1996b, Geometric and Topological Consistency of Spatial Data, *in the Proceedings of the First International Conference on Geocomputation*, Leeds, UK, 17-19 September 1996, pages 830-842.

Ubeda T., 1996c, Les modèles de données et le contrôle de la cohérence dans les bases de données géographiques, Technical report, RR96-13, Laboratoire d'Ingénierie des Systèmes d'Information, INSA de Lyon, 71 pages.