Geoinformatics 2004
Proc. 12th Int. Conf. on Geoinformatics − Geospatial Information Research: Bridging the Pacific and Atlantic
University of Gävle, Sweden, 7-9 June 2004

FROM THE SCHEMA MATCHING TO THE INTEGRATION OF UPDATING INFORMATION INTO USER GEOGRAPHIC DATABASES

Arnaud Braun
Institut Géographique National / Laboratoire COGIT, 2-4 avenue Pasteur, 94165 Saint Mandé cedex, France
Arnaud.Braun@ign.fr, Tel: +33.1.43.98.80.00, Fax: +33.1.43.98.81.71

Abstract

This article presents the results of a study on the definition of models and functionalities of tools for the integration of updating information into geographic databases. This updating information is delivered by a geographic information producer (such as a National Mapping Agency). These tools should be generic, i.e. independent of geographic databases and GIS software. This can be reach with a rigorous schema matching between user and producer databases schemas. With such a schema matching, a process with four stages is proposed: scheduling and grouping, filtering, integration of updates and management of conflicts. As a conclusion, implementation of a portable prototype is proposed.

INTRODUCTION

Propagating updates between geographic databases with different scales is a crucial problem. Indeed, users of geographic data sets (provided by geographic information producers such as National Mapping Agencies) clearly need updates from producers in order to have the most realistic image of geographic reality. These users are any organizations dealing with geographic data (governmental agencies, local institutions, private companies, etc.). They often add their own knowledge in their GIS. Integrating updates provided by a producer into user data sets must allow for the preservation of this added knowledge, in order to avoid loss of information or inconsistent states of the geographic database. Producers themselves may have similar preoccupations if they want to propagate updates from their reference data sets to their derived products (e.g. cartographic products or other databases with different scales).

The paper first presents the context and objectives of this study and discusses the formats of updating information. Schema matching step is detailed in a second part. The article then focuses on the integration mechanism of the updates. Points for a portable implementation and for a potential deployment as a “Web Services Architecture” are presented as a conclusion.

RELATED WORK AND OBJECTIVES

Existing solutions

A generic mechanism for integrating and propagating updates between geographic databases has been proposed in Badard and Lemarié (1999) and in Badard (2000). A prototype has been developed at the COGIT Laboratory of IGN (the French National Mapping Agency) with the object-oriented geographic database management system GeO2 (Raynal et al., 1995). It has been tested on geographic databases whose schemas and scales are close. Input information is updating information, delivered with different kinds of spatiotemporal evolutions: creation, deletion, thematic attribute modification, geometrical modification, merging, splitting, aggregation. There are two kinds of structures of delivery: differential datasets or evolution messages. Differential datasets allow for an updating information in form of references on couples old object / new object. Nowadays, such datasets are computed in IGN and delivered in Shapefile format. As for evolution messages, they do not require any delivery of objects. Only the description of necessary transformations between the old objects and the new objects, is delivered. A structure based on GML is proposed in Badard and Richard (2001). The solution recommended in this article is applicable for both updating structures.

Results of these experiments have been embedded in an industrial project at the IGN: see Jahard et al. (2003). This project, which is now coming to maturity, allows for a very significant saving of time for updating tasks: 60 hours are required with this new automatic process instead of 300 hours with the former process (with a visual comparison between two data sets), for updating a 1:100.000 cartographic database with a differential dataset.

Limits of these solutions

Such experiments in both research and production are very rich, but come up against different kinds of problems. They are not generic neither portable: solutions focus on the update of specific products and are dependant on GIS software. They have been developed with non-open or proprietary systems (operating system, programming language, GIS platform). Migrating them to another systems, creating a new “updating line” in a production context for other products and other GIS platforms, or using them outside IGN may require an almost complete remodelling of methods and rewriting of tools. Such an reengineering may be more or less time consuming and depends on the differences between database structures and GIS software.

Objectives

The objective of our study was to define models and functionalities of a set of tools that allow for the integration of updating information delivered by a geographic information producer into user geographic databases. The solution should be generic and portable:

- It has to be independent of database structures (both user and producer) for being used by any organization. Scale gap between databases, and use of both geographic and cartographic databases must be allowed.
- It has to be to a maximum independent of GIS platforms to avoid problems of reengineering.
- It has to be independent of operating systems and hardware platforms in order to be potentially accessible via the web.

Global process

The global process is illustrated in Figure 1. Starting from differential datasets or evolution messages, user database has to be updated. This can be generically performed with schema matching.

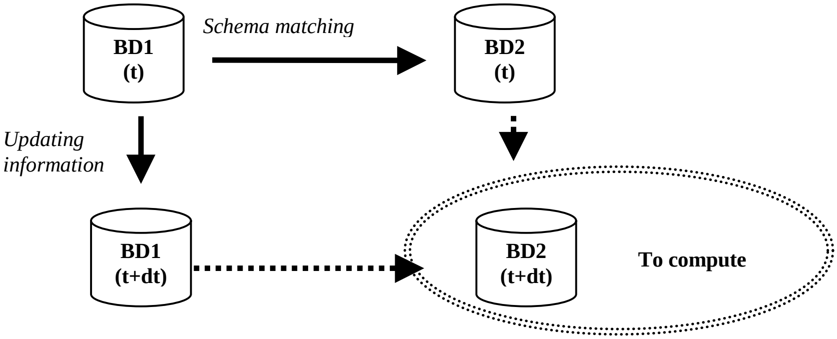

Figure 1: Global process (BD 1 = producer database; BD2 = user database).

SCHEMA MATCHING

Finding correspondences between database models is called schema matching. It is a major topic of ongoing researches in the database community, see e.g. Pottinger and Bernstein (2003). The problem is: how to find classes, attributes and relations representing the same features of the real world in both user and producer schemas? Such a schema matching may have different uses, not only integration of updates (the subject of this article) but also derivation of databases or control of consistency. For our integration tasks, updating information is contained in a database called “source database”, and integration of updates have to be executed in the second database called “target database”.

Inter-schema correspondences

To identify and qualify relationships between models, we propose the use of the grammar for the “Inter-schema Correspondences Assertions (ICA)” defined in Devogele et al. (1998), which have been tested on IGN’s geographic databases. This grammar is simple, precise, and allows for the expression of correspondences between classes and attributes from database schemas with different levels of constraints. Assuming S_i and S_j denote the schemas of two databases, the general format for an ICA includes four clauses:

ICA S_i.item$_i$ *correspondence-relationship Sj*.item$_\mathrm{j}$
SDM instance-matching predicate
WCA attribute-correspondences
WCG geometry-correspondence

where item$_i$ and item$_\mathrm{j}$ are names of a schema element (or expressions involving schema elements) in respectively S_i and *Sj*, and *correspondence-relationship* is one of the usual mathematical relationships between sets: $\equiv$, $\subset$, $\subseteq$, $\supset$, $\supseteq$, $\cap$, $\neq$.

The clause “ICA” answers the question: what are the related schema elements ? In the simplest case, instances of a given type (e.g. item$_i$) in one database are one to one related to instances of a given type (e.g. item$_\mathrm{j}$) in the other database. More frequent is the case where related sets of real world things are not exactly equivalent. Mapping may have different cardinality: 1-1, 1:N, N:1 or N:M. The kind of relationship (equivalence, inclusion, insertion, etc.) answers the question: how are their population related ? In this study we have extended this grammar in order to express classical database derivation operations such as selection, joint, projection. Indeed, this grammar was initially defined for expressing correspondences between independent databases. Our process must be able to deal with a target databases directly derived from a source database, and such a derivation information must also be expressed.

The clause “SDM” (Spatial Data Matching) answers the question: how are corresponding instances identified? When it is possible, this can be done with value-based identifiers (primary-key) if such identifiers exist in the database and if the target database is derived from the source database, with preservation of identifiers. In other case, when both databases are independent, corresponding instances are retrieved with geographic data matching. These geographic data matching tools rely on methods explained in Devogele (1997) and Badard (2000).

The clauses “WCA” (With Corresponding Attributes) and “WCG” (With Corresponding Geometry) answer the question: how are representations related, in terms of either their thematic attributes or their geometry ? A WCA clause can be expressed with different levels of constraints: attribute values of the target database can be directly derived from values of the source database, or they can not be directly derived but values can be selected in a restricted set of values depending of values in the source database, thanks to correspondences functions, as shown in the example below. This allows for the preservation of the database consistency for non-geometric attributes.

A WCG clause is used whenever it is possible to specify how the geometry in the target database is resulting from a derivation process performed on the geometry in the source database. If the target database is a cartographic database, derived from a producer database, such a derivation is performed to increase cartographic legibility. Some of geometric process may be automatic, others are interactive. Examples of such process are: Douglas-Peucker filtering, smoothing, merging, splitting, coordinate transformation, etc. Such a rule can be simulated, e.g. if no process of derivation is known: a simple morphing algorithm mimics this process.

Example

The following example has been established and tested with real databases from IGN. It deals with road section classes of two databases (the target database BD2 is BDCarto©, a ~1:50.000 geographic database, and the source database BD1 is GeoRoute©, a ~1:10.000 geographic database dedicated to in-car navigation). These two databases are fully independent, so instances correspondences are established by the way of geographic data matching process. An example of correspondence function between thematic attributes is shown. Attribute values can not always be directly derived, but constraints on them can be established. Finally, the last clause shows that geometries of target database may be derived from geometries of source database by a morphing algorithm: this is a simulation of a derivation process.

ICA : *BD2.RoadSection ⊂ (SET[1:N] BD1.RoadSection WHERE attribut1 != 'value1')*

SDM : *BD2.RoadSection.geometry (linestring) ± BD1.RoadSection.geometry (linestring)*

WCA : *BD2.RoadSection.vocation = f1 (BD1.RoadSection.vocation, BD1.RoadSection.nature)*

WCG : *BD2.RoadSection.geometry (linestring) = morphing (BD1.RoadSection.geometry (linestring))*

f1 :

BD1.RoadSection.vocation ("principal") ⇒ *BD2.RoadSection.vocation* ("type autoroutier")

BD1.RoadSection.vocation ("primaire") ⇒ *BD2.RoadSection.vocation* ("type autoroutier")
or *BD2.RoadSection.vocation* ("liaison principale")

BD1.RoadSection.vocation ("secondaire") ⇒ *BD2.RoadSection.vocation* ("liaison principale")
or *BD2.RoadSection.vocation* ("liaison régionale")
or *BD2.RoadSection.vocation* ("liaison locale")
BD1.RoadSection.vocation ("desserte") ⇒ *BD2.RoadSection.vocation* ("liaison locale")
BD1.RoadSection.nature ("autoroute") ⇒ *BD2.RoadSection.vocation* ("type autoroutier")
BD1.RoadSection.nature ("quasi-autoroute") ⇒ *BD2.RoadSection.vocation* ("type autoroutier")
BD1.RoadSection.nature ("chemin") ⇒ *BD2.RoadSection.vocation* ("liaison locale")

Object-oriented model

In order to store these correspondences relationships, we have proposed an object-oriented model. Indeed, these relationships will be used in the integration process and must be retrieved. An extract of this model is provided in Figure 2 with UML formalism: there are classes for representing ICA (named ACI in French), SDM (named AIC), WCA (named AAC) and WCG (named "RegleGeometrique"). This model allows for a complete navigation between features and attributes of both databases and for a complete representation of ICA concepts. XML schemas based on GML 2.0 have been defined to implement this model. The use of XML ensure the portability of this information. The use of GML allows for a standard way to describe databases schemas.

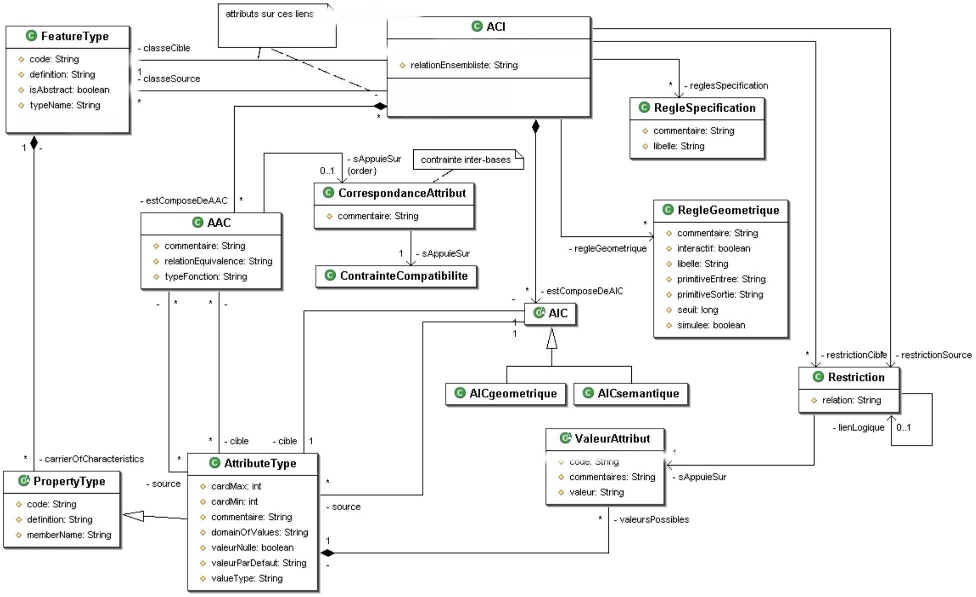

Figure 2: Extract of object-oriented model for schema matching (UML formalism).

INTEGRATION OF UPDATING INFORMATION

General process of integration

The overall process for the integration of updates is presented in Figure 3. The process is decomposed in four stages: scheduling and grouping, filtering, integration of updates and management of conflicts. All information for the first three steps (criteria, algorithms choice, thresholds) as well as schema matching data are stored in a rule base in form of XML files according to XML Schemas. The use of XML ensure the portability of this information, and its capability to be processed with standard solutions such as DOM or SAX.

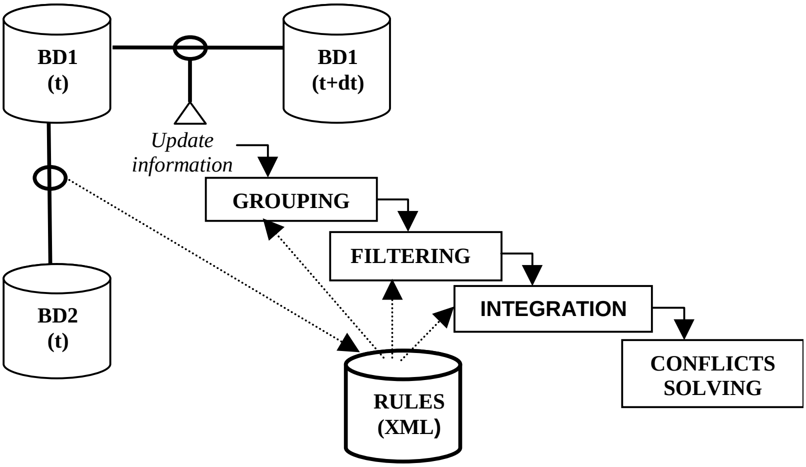

Figure 3: General process of integration.

Scheduling and grouping

The purpose is to decrease the number of potential updating conflicts that will appear during the integration step. Scheduling is based on the type of geographic entities (e.g. road network, buildings) and on the kind of updates: it is better to first integrate the deletions, otherwise intersection conflicts with destroyed objects may occur. Then grouping of updates is done according to geometrical criteria. For example, networks have to be formed again to integrate all new connected road sections as a single update, otherwise a network may be declared in a situation of conflict if a constraint imposes that a network section must be connected to an another network section. Schema matching is not really used during this stage. Criteria for grouping and scheduling (thresholds, algorithms) are stored in the rule base.

Filtering

The purpose is to determine updates relevant for the target database, according to both semantic and geometrical criteria. Schema matching is here fully used. Indeed, only classes of the target database that are involved in ICA clauses are retained; only thematic attributes that are involved in WCA clauses are retained. And only instances (expect for creation) that are involved in instance matching are used. As for geometry, a geometrical filtering is performed according to the resolution of the target database which can be very less than the resolution of the source database. If the geometry of a new object is not significant for the resolution of the target database, this object is not retained. Size (length, area) of new object, and Haussdorff distance between the old and the new object in case of geometrical modification, are used as criteria. With such a filtering, experiments show that about 30 % of updates may be eliminated. Criteria for filtering (thresholds, algorithms) are also stored in the XML rule base.

Integration

To achieve this step, we have proposed many algorithms, which depend on the kind of object (point, line, area), the kind of update (deletion, creation, splitting, merging, aggregation, thematic attribute modification, or geometrical modification), and the cardinality of the correspondence relationship between the producer and the user data sets (i.e. 1-to-1, 1-to-N, N-to-1, or N-to-M) defined in the ICA clauses of correspondences relationships. Examples of such algorithms for geometrical modifications are shown in Table 1. Such tables have also been designed for creation, deletion, merging, splitting, aggregation. The underlying idea is to have a family of “pluggable” algorithms that user can select according to the specifications of its database. The selection of these algorithms with associated thresholds is performed thanks to the XML rule base. WCG clause allows for processing a geometric algorithm. As for thematic attributes modification, thanks to WCA clauses, and correspondences functions, new thematic attributes of target database can be selected or at least some coherent indication can be proposed.

Table 1: Algorithms for geometrical modifications.

Type of entities	Cardinality of relationship		
Source / Target	1 –1	N – 1 (fusion)	1 – N or N – M
Point / point	Displacement of target point with the same displacement of source point, then geometric rule according to WCG clause.	Idem as 1-1 case by computing centroïd of source points, then geometric rule according to WCG clause.	INDECISION CONFLICT
Linestring / point	Displacement of target point with the same displacement of centroïd of source linestring, then geometric rule according to WCG clause.	Idem as 1-1 case by computing centroïd of source linestrings, then geometric rule according to WCG clause.	
Polygon / point	Displacement of target point with the same displacement of centroïd of source polygon, then geometric rule according to WCG clause.	Idem as 1-1 case by computing centroïd of source polygons, then geometric rule according to WCG clause.	
Linestring / linestring	Geometric rule according to WCG clause.	Idem as 1-1 case by merging source linestrings.	
Polygon / polygon	Geometric rule according to WCG clause.	Idem as 1-1 case by merging source polygons.	

Such an integration may generate some conflicts, which have been classified in two categories: indecision conflicts (i.e. when no solution can be proposed by the process) or violation conflicts (i.e. when an integrity constraint is violated, e.g. non-continuous network, non permitted intersection between objects, etc.). Algorithms for verification of constraints are also pluggable and integrated in the XML rule base.

Management and resolution of conflicts

An interactive interface that allows for resolution of conflicts previously identified (indecision conflicts and conflicts of violation of constraint) has to be designed. This interface is called at the end of the integration step of each theme. The definition of such an interface requires complementary studies. Such an interface must allow the guided solving of the conflicts and relies on a semiology of the updates and conflicts, in order to facilitate the understanding of the conflicts and how to solve them.

CONCLUSION AND OUTLOOKS

Contribution

This article presents a process for integration of updating information into user geographic databases. It uses schema matching information to be generic, i.e. independent of databases structures. Using an existing grammar, schema matching information can be stored in XML files thanks to the definition of an object-oriented model. We have proposed a mechanism, using schema matching data and pluggable algorithms whose choice is made using an XML rule base. We have designed all XML schemas for schema matching storing and integration process rules storing. The use of XML allows for portability, i.e. the independence of hardware and operating systems, to be potentially accessible via the web.

Toward a prototype

We propose an implementation of models and functionalities detailed in this article in the new interoperable platform of the COGIT laboratory, OXYGENE: see Badard and Braun (2003). With portable technologies such as Java and XML, natively embedded in this platform, portability of the prototype we aim to develop is allowed. The first three steps of the proposed mechanism may be entirely independent of any GIS platform: grouping, filtering or integration. Only the management and resolution of conflicts may deal with human interaction and can be embedded either in a GIS platform or in a self-coded interface. This independence of the modules is possible with a rigorous definition of constraints and conflicts that requires complementary studies. With these technologic recommendations (Java, XML, independence of GIS platforms), the definition of a full decentralized architecture would be possible for the implementation of “updating web services”.

REFERENCES

Badard, T., 2000: *Propagation des mises à jour dans les bases de données géographiques multi-représentations par analyse des changements géographiques*. Ph-D. Thesis. Université de Marne-la-Vallée.

Badard, T. and Braun, A., 2003: OXYGENE: an Open Framework for the Deployment of Geographic Web Services. In: *Proceedings of the 21st International Cartographic Conference (ICC).* Durban, South Africa, 994-1004.

Badard, T. and Lemarié, C., 1999: Propagating updates between geographic databases with different scales. In: *Innovations in GIS VII: GeoComputation*, Atkinson, P. and Martin, D. (Eds.), Taylor and Francis, London, Chapter 10.

Badard, T. and Richard, D., 2001: Using XML for the Exchange of Updating Information Between Geographical Information Systems. *Computer, Environment and Urban Systems (CEUS)* 25, 17-31.

Devogele, T., 1997: *Processus d’intégration et d’appariement des bases de données géographiques, application à une base de données routières multi-échelles*. Ph-D. Thesis. Université de Versailles.

Devogele, T., Parent, C. and Spaccapietra, S., 1998: On Spatial Database Integration. *International Journal of Geographical information Science (IJGIS)* 12(4), 335-352.

Jahard, Y., Lemarié, C. and Lecordix, F., 2003: The implementation of new technology to automate map generalisation and incremental updating processes. In: *Proceedings of the 21st International Cartographic Conference (ICC)*. Durban, South Africa, 1449-1459.

Pottinger, R. and Bernstein, P., 2003: Merging models based on given correspondences. In: *Proceedings of the 29th International Conference on Very Large Databases (VLDB),* Berlin, Germany, 862-873.

Raynal, L., David, B. and Schorter, G., 1995: Building an OOGIS Prototype: Experiments with GeO2. In: *Proceedings of ACSM/ASPRS Annual Convention and Exposition (AutoCarto12).* Charlotte, North Carolina, USA, volume 4, 137-146.