MATCHING CARTOGRAPHIC OBJECTS IN SPATIAL DATABASES

Daniela Mantel, Udo Lipeck

Database Group, Information Systems Institute, University of Hannover, Germany

KEY WORDS: Federated Databases, Matching Algorithms, Integration, Resolution

ABSTRACT:

Nowadays cartographic objects of different resolutions are hold in different coexisting databases. This implies an extensive amount of work for updating an object in all resolutions. One way to reduce this is to build a multi-resolution database which holds and links different representations of the real-world objects and allows to automatically pass updates to all linked representations (requiring algorithms for propagation of updates). In order to ensure the autonomy of applications on the local databases, we propose the architecture of a federated database for such a multi-resolution database.

A main requirement for setting up a multi-resolution database is to "identify" the different representations of real-world objects. Therefore we propose a multistage procedure as follows:

- Semantic classification: Identify the sets of objects to compare with one another
- Compute geometrically possible matchings within these classes
- Postprocessing: Automatically select correct matchings from the set of possible matchings by applying prepared rules
- Manual select correct matchings for the remaining possible matchings, which were not automatically detected as correct or incorrect

We present a framework for the needed semantic classification, a concept for rule-based selection as well as an algorithm to compute possible matchings.

We have enhanced the formerly known buffer growing algorithm for computation of possible matchings and implemented it in PL/SQL for use in spatial databases based on Oracle9i (with the spatial data cartridge). The enclosing object matching process is supported by a graphical user interface utilizing stored database procedures for the mentioned steps and rules.

KURZFASSUNG:

Kartographische Objekte verschiedener Maßstäbe werden in unterschiedlichen voneinander unabhängigen Datenbanken gehalten. Dies führt zu einem hohen Aufwand in der Fortführung. Um diesen Aufwand zu reduzieren, wird die automatische Übertragung von Veränderungen von einem Maßstab in den nächsten in Betracht gezogen. Voraussetzung dafür ist, dass die Datenbestände miteinander verknüpft sind. Dies kann in einer Multi-Resolution-Database (MRDB) abgebildet werden, die sowohl die unterschiedlichen Datenbestände als auch die Verknüpfungen zwischen den Objekten, die das gleiche Real-Welt-Objekt repräsentieren, speichert. Um hier die Autonomie der zugrunde liegenden Datenbestände zu gewährleisten, schlagen wir als Architektur einer solchen MRDB eine föderierte Datenbank vor.

Beim Aufbau der MRDB ist die Objektidentifikation, das heißt das Bestimmen der Objektmengen, die jeweils das gleiche Real-Welt Objekt beschreiben, ein Hauptproblem. Hierfür kann ein schrittweises Vorgehen gewählt werden:

- Semantische Klassifikation, das heißt Bestimmung der jeweils zu vergleichenden Objektmengen
- Geometrische Ermittlung von möglichen Zuordnungen innerhalb dieser Mengen
- Regelbasierte Auswahl von richtigen Zuordnungen aus der Menge der möglichen Zuordnungen
- Manuelle Auswahl für die möglichen Zuordnungen, die nicht automatisch bestätigt oder verworfen werden konnten.

In diesem Artikel stellen wir ein Vorgehen für die benötigte semantische Klassifikation sowie einen Algorithmus für die Ermittlung der möglichen Zuordnungen und ein Konzept für die regelbasierte Auswahl vor.

Wir haben den bekannten Buffer Growing Algorithmus zur Ermittlung möglicher Zuordnungen auf symmetrische Matching-Situationen und auf die mengenorientierte Verarbeitung in einer Datenbank angepasst und ihn in PL/SQL zur Verwendung in Oracle 9i (mit räumlicher Erweiterung) implementiert. Der gesamte Prozess der Objektidentifikation wird durch eine graphische Benutzeroberfläche unterstützt, die mit Prozeduren der Datenbank arbeitet.

1. INTRODUCTION

1.1 Motivation

For many cartographic datamodels there are multiple databases each representing part of earths surface in a specified resolution. For example for the german ATKIS-model there exist independently mapped datasets for the resolutions 1:25.000, 1:250.000 and 1:1.000.000. The necessity to update these datasets causes an extensive amount of work, because every single dataset has to be adjusted manually.

One way to reduce this is to automate part of the work, that is to update only one resolution, the finest, and then propagate the changes to all other resolutions. A prerequisite for this procedure is, that the access from one object to all corresponding representations of the same real world object is possible. A concept for such a data structure is the multi-resolution database.

Because cartographic objects often lack an explicit (and to the represented real-world object related) identifier, a main issue in the process of setting up a multi-resolution database is to identify objects which represent the same real-world object.

This paper describes an architecture for a multi-resolution database, that is based on the paradigm for federated databases, and on a framework for the computation of object matchings.

1.2 Related Work

In (Walter, 1997) the buffer growing algorithm which we use for parts of the geometric matching is described. That paper also makes a suggestion for a selection process based on relational quality of the set of chosen matchings and develops an algorithm to find the "best" set of matchings.

(Sester et al., 1999) gives an overview of different approaches for finding links between representations of real world objects.

(Kleiner et al., 2001) develops a system for the storage of geographic objects in object-relational databases, which we use for the component databases to be integrated in the multi-resolution database.

(Conrad, 1997) gives an overview of the paradigm of federated databases and methods to generate it. In particular different methods for conflict resolution during integration of database schemas are described.

2. STRUCTURE OF THE MULTI-RESOLUTION DATABASE

2.1 Architecture and system structure

To maintain the cartographic quality of the different datasets, it is useful to keep the original databases and separate the needed integration from them. A reference architecture for such purposes is the federated database (see Conrad, 1997). This architecture guarantees a maximum of autonomy for the so-called component databases, i. e. the databases to be integrated, while enabling an integrated access to them for global applications.

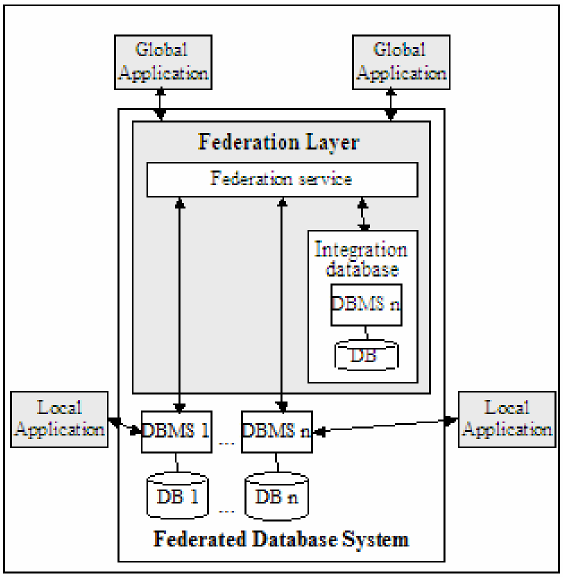

Figure 1: Architecture of a federated database system

The principle structure of such an architecture is shown in figure 1: The (unchanged) component databases still support their local applications. They are integrated via a federation layer, which offers the global access to them. The federation layer maintains the links between correponding objects and holds the meta data for the access and for processes, for example for matching and generalization. Therefore the database for the federation layer consists of the mentioned parts as shown in figure 2.

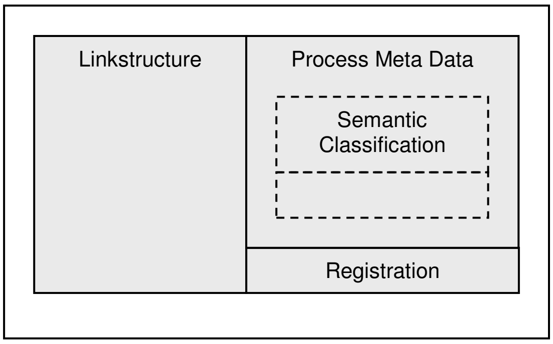

Figure 2: The integration database

2.2 Link structure

Real world objects often are represented as combinations of more than one database object. For example street sections between crossings can be broken into multiple segments with respect to some attribute, say name of the street or their width. If these criteria are different in the datasets to be integrated, because of different datamodels or differing tresholds for data capture or any other reason, the real world object may be represented in sets of database objects, which cannot be matched one by one. In figure 5 such a situation is shown. The street section is represented by three objects in database A and two objects in database B and there is no correspondence between a pair of objects from database A and database B.

Therefore a structure for storing matchings has to deal with matchings of cardinality many-to-many as well as cardinalities one-to-one and one-to-many. Figure 3 shows a schema that satisfies this requirement. The sets of objects representing a real-world object are modeled as aggregated objects, which are associated via "Matching" to the aggregated object representing the same real world object in the database to be linked. To improve the performance of spatial queries the aggregated geometry for each aggregated object is stored. The topological relations between the single objects are modeled in the class "Relation".

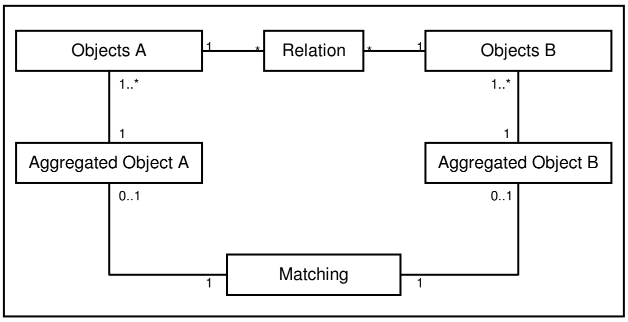

Figure 3: Link structure

3. MATCHING PROCESS

To find the links between corresponding objects representing the same real world object we propose a stepwise process as follows. First the input sets for the geometric algorithms should be as small as possible without losing quality of results. Therefore the first step is to divide the object sets into sets of comparable object types, that is to accomplish a semantic classification on the object sets in the component databases. The details of this step are described in paragraph 3.1.

The next step is to find the geometrically possible matchings, that is the pairs of object sets which are geometrically likely to represent the same real world object, within the comparable object types. An algorithm for this purpose is detailed in paragraph 3.2.

The mentioned algorithm computes more than just the "correct" links, so that subsets of "confirmed" (which means correct) and "discarded" matchings need to be selected from the result set. This should be widely automated, as suggested in paragraph 3.3.

After the automatic selection procedures there will remain some matchings, which could not automatically be confirmed or discarded. For such cases of doubt an interface is needed which provides an operator with tools to manually handle this set. The requirements for this interface are presented in paragraph 3.3.3.

3.1 Semantic classification

To reduce the necessary amount of computations filtering should be done which defines the input for the following matching algorithm. Such a filter should separate all object classes which can never represent the same real-world object, but must not exclude any possible n:m matching. Consider for example the two database schemas in figure 4 a). In database A traffic routes are modelled in the classes highway, street and alley. The differentiation between streets and alleys is made by the importance of the roads for transit traffic. In database B traffic routes are modeled in the classes street and alley, whereas the differentiation is made by means of paving, that is streets have tramac, alleys not. An algorithm for determination of possible matchings should obviously compare both streets and highways of database A with the streets of database B as well as the alleys of database A with the alleys of database B. Furthermore the comparison must be drawn between the streets of database A and the alleys of database B. In figure 4 b) all direct comparisons are shown as associations between the classes.

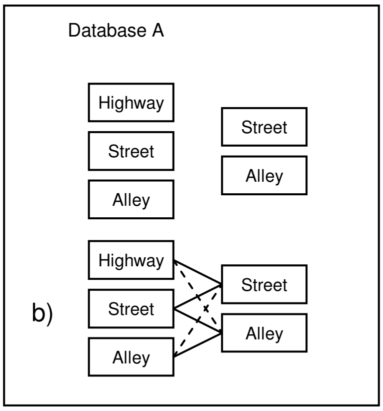

Figure 4: Semantically corresponding classes

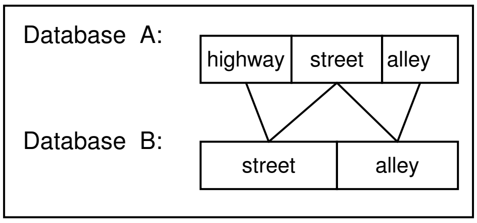

Figure 5

Beyond this, if a situation like in figure 5 occurs on the object level, one cannot set aside the indirect associations (shown with dashed lines in figure 4 b)) between highways in database A and alleys in database B. Therefore the filter should in the first step only separate such classes, that are not even indirectly associated with one another, e. g. Such aggregated streets from (maybe aggregated) railroad lines. We call the result of this step coarse class matching, the considered object sets in the databases coarse compare sets.

In the next step, it has to be examined, wether there is an attribute in both coarse compare sets dividing these sets into disjoint comparable sets, in our example say an attribute which says if the traffic route is inside an urban area or out of town. The classes within the coarse compare sets can be divided into smaller, disjoint sets. And the corresponding coarse compare sets and the coarse class matching can be divided by direct derivation from these without the risk to lose an essential input for the matching algorithm. We call such characterizing attributes "partitioning attributes". When all partitioning attributes are applied, the resulting sets are called (fine) class matching and compare sets respectively. The subsets of classes forming the compare sets are called object types.

The description of the semantic classification is stored in the integration database according to the schema in figure 6.

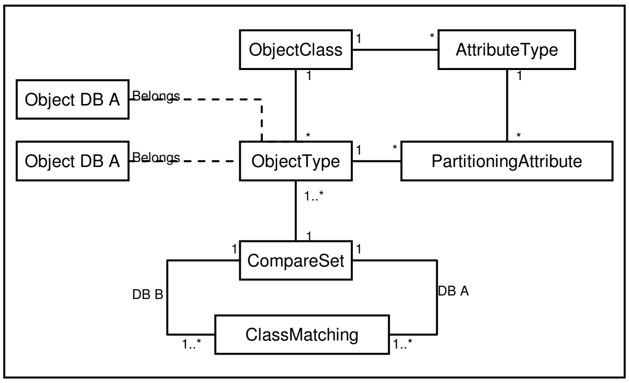

Figure 6: Schema for semantic integration

3.2 Buffer growing

An algorithm to compute possible matchings between line-objects is "buffer growing" (see Walter, 1997). It acts on the assumption that representations of the same real-world object have similar locations (after possibly necessary coordinate transformations). In the cited paper the algorithm is described in a purely iterative way prioritising one of the two datasets that should be matched. We have made it symmetric for use with respect to two datasets on a par, and we have adapted it for the more set oriented process in a database system.

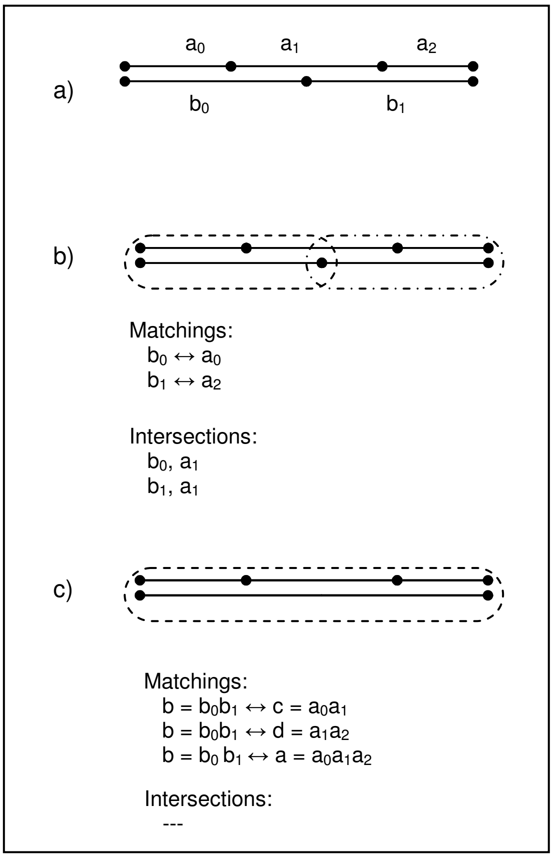
a)
a_0
a_1
a_2
b_0
b_1
b)
Matchings:
$b_0 \leftrightarrow a_0$
$b_1 \leftrightarrow a_2$
Intersections:
b_0, a_1
b_1, a_1
c)
Matchings:
$b = b_0b_1 \leftrightarrow c = a_0a_1$
$b = b_0b_1 \leftrightarrow d = a_1a_2$
$b = b_0 b_1 \leftrightarrow a = a_0a_1a_2$
Intersections:

Figure 7: Buffer Growing

The algorithm executes as follows: A buffer is built around every object, and all objects of the other dataset which are totally inside this buffer as well as all geometrically possible aggregations of them are identified as possible matching partners as shown in figure 7 b) for the objects b_0 and b_1. If two buffers are intersecting with the same object, like a_1 intersecting the buffers around b_0 and b_1 in our example, the corresponding objects of the buffers are aggregated (if possible) and the aggregated object is treated like a single object, that is, another buffer is built around this new object to find possible matching partners, for example in figure 7 c) the matchings containing object b. To avoid duplicate aggregated objects, new aggregations are only built, if they contain the intersecting object. Matchings, which are just extensions of another matching (e. g. $b_0b_1 \leftrightarrow a_0$ is an extension of $b_0 \leftrightarrow a_0$ in our example) are not stored as a possible matching.

In this way the algorithm computes all possible matchings of cardinalities one-to-one, one-to-many and many-to-many. Of course, it depends on parameters like the buffer distance which have to be tuned for the datasets at hands.

3.3 Selection

The set of possible matchings does not only contain correct matchings, but some incorrect suggestions too. Therefore the elements of the set have to be subdivided into confirmed and discarded matchings as mentioned above.

3.3.1 Conflicts

If the same object is part of two or more aggregated objects involved in different possible matchings, no two of these matchings can be simultaneously correct. The matchings are said to be in conflict with one another. Therefore, if a possible matching is confirmed, all possible matchings, that are conflicting with it, have to be discarded.

On the other hand, if a possible matching is "good enough", that is, it fulfills all quality criteria (see below), and is not conflicting with any other still possible matching, it can be confirmed.

3.3.2 Automatic Selection

Automatic selection of matchings, that is confirming or discarding of them, can be controled by rules. A special type of rules are rules for checking quality criteria. Quality criteria are measures for the resemblance of a single aspect of the matched objects, e. g. length of matched lines, similarity of names etc..

There are different approaches to use quality criteria for automatic selection of correct matchings.

One can compute all the measures of each criterion for all possible matchings and then compute the "best combination" of matching, that is the combination with the highest sum of measures. This problem is equivalent to searching the best complete subgraph (clique) in a graph with weighted vertices.

Another approach is to define a treshold value and discard all matchings, with quality measure below the treshold. After discarding, all remaining possible matchings, which are not in conflict with any other any more, can be confirmed.

The latter approach can be refined to an iterative method, by starting the selection with a high treshold value and then decrease it stepwise until a minimum treshold value is reached. In every step all matchings with quality measure below the treshold are "temporarily" discarded and non-conflicting matchings are confirmed. When confirming a matching, all temporarily discarded matchings, which are in conflict with it, have to be definitely discarded.

For line objects this iterative approach leads to good results for the criterion “similarity of length”, which means that a treshold is defined for the maximum of the two quotients of the length.

3.3.3 Manual Selection

The automatic selection procedure leaves a set of possible matchings which cannot automatically be confirmed or discarded. These are, for example, pairs of matchings, which hold the given set of quality tresholds and are in conflict with each other. In such cases the decision for confirming or discarding must be left to a human operator.

The operator must be provided with an interface, which helps him making a decision. Therefore a graphical user interface is needed, which shows the uncertain matchings in their context and lets the user confirm or discard.

We have implemented an extension for the visualizer GISVisual, which was formerly developed at our institute, which provides the user with these features and an interface for administration of the federated database. The interface provides firstly a graphical user interface for capturing and changing the needed meta data and parameters, for example for the semantic classification. Then there is the possibility to register and parameterize procedures for finding possible matchings (as an alternative to the buffer growing) as well as for the selection procedures. These are procedures implemented in PL/SQL.

For the manual selection the functions for marking pairs of objects, in this context the matching pairs, and calling database procedures on this pair, were implemented. The operator therefore can choose one or more pairs and afterwards discard or confirm them with respect to the conflict rules. If configured by the operator, the process of confirmation of non-conflicting matchings, will start after each manual confirmation.

4. FUTURE WORK

Now, that we have a framework for generating a multi-resolution database and some methods to match line objects, we are focussing on tuning the matching process and augmenting the degree of automation in the selection process, which means to experiment with different parameterizations for the existing procedures as well as developing new procedures.

Another focus has to be set on the development of region matching algorithm respectively the integration of exiisting ones.

5. REFERENCES

Conrad, S., 1997. *Föderierte Datenbanksysteme.* Springer Verlag, Berlin.

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

Kleiner, C., Lipeck, U.W., 2001. *Enabling Geographic Data with Object-Relational Databases.* In: A. Heuer et al., Datenbanksysteme in Büro, Technik und Wissenschaft – 9. GI-Fachtagung BTW 2001, Springer Verlag, Berlin, pp. 127-143.

Lipeck, U.W., Mantel, D., 2004. *Datenbankgestütztes Matching von Kartenobjekten*. To appear in: Mitteilungen des Bundesamtes für Kartographie und Geodäsie, Bundesamt für Kartographie und Geodäsie, Frankfurt am Main.

Mantel, D., 2002. Konzeption eines Föderierungsdienstes für geographische Datenbanken. Master Thesis, University of Hannover, Germany.

Sester, M., Anders, K.-H., Walter, V., 1999. Linking Objects of Different Spatial Datasets by Integration and Aggregation. *GeoInformatica*, 2(4), pp. 335-358.

Sester, M., 2000. Maßstabsabhängige Darstellungen in digitalen räumlichen Datenbeständen. Habilitation Thesis, University of Stuttgart, Germany.

Tiedge M., Lipeck, U. and Mantel, D., 2004. *Design of a Database System for Linking Geoscientific Data*. Geotechnologien Science Report "Information Systems in Earth Management", No. 4, Koordinierungsbüro Geotechnologien, Potsdam, 2004, pp. 83-87.

Walter, V., 1997. Zuordnung von raumbezogenen Daten – am Beispiel der Datenmodelle ATKIS und GDF. Ph.D. Thesis, University of Stuttgart, Germany.

6. ACKNOWLEDGEMENTS

This work is funded by the German Federal Agency for Cartography and Geodesy as part of the WiPKA (Wissensbasierter Photogrammetrisch-Kartographischer Arbeitsplatz) project and by the Federal Ministry for Education and Research (BMBF) and the German Council (DFG) as part of the Geotechnologien project under contract no. 30F0374A.