To link to this article : DOI:10.1080/13658810701791949
URL : http://dx.doi.org/10.1080/13658810701791949

A data-mining approach for assessing consistency between multiple representations in spatial databases

D. SHEEREN*†‡, S. MUSTIÈRE† and J.-D. ZUCKER§

†COGIT Laboratory, Institut Géographique National (IGN), Paris, France
‡LSIIT, Data Mining Group, UMR 7005—CNRS/ULP Strasbourg, France
§GEODES, UR 079, IRD, Bondy, France

When different spatial databases are combined, an important issue is the identification of inconsistencies between data. Quite often, representations of the same geographical entities in databases are different and reflect different points of view. In order to fully take advantage of these differences when object instances are associated, a key issue is to determine whether the differences are normal, i.e. explained by the database specifications, or if they are due to erroneous or outdated data in one database. In this paper, we propose a knowledge-based approach to partially automate the consistency assessment between multiple representations of data. The inconsistency detection is viewed as a knowledge-acquisition problem, the source of knowledge being the data. The consistency assessment is carried out by applying a proposed method called MECO. This method is itself parameterized by some domain knowledge obtained from a second method called MACO. MACO supports two approaches (direct or indirect) to perform the knowledge acquisition using data-mining techniques. In particular, a supervised learning approach is defined to automate the knowledge acquisition so as to drastically reduce the human-domain expert's work. Thanks to this approach, the knowledge-acquisition process is sped up and less expert-dependent. Training examples are obtained automatically upon completion of the spatial data matching. Knowledge extraction from data following this bottom-up approach is particularly useful, since the database specifications are generally complex, difficult to analyse, and manually encoded. Such a data-driven process also sheds some light on the gap between textual specifications and those actually used to produce the data. The methodology is illustrated and experimentally validated by comparing geometrical representations and attribute values of different vector spatial databases. The advantages and limits of such partially automatic approaches are discussed, and some future works are suggested.

Keywords: Data mining; Inconsistency; Integration; Metadata; Multiple representation; Spatial data matching

1. Introduction

In the present era of information, the co-existence of multiple representations of the same phenomena has become usual. This is a general observation which is particularly true in the field of geographical information. Several technical reasons

*Corresponding author. Email: david.sheeren@ensat.fr

can explain this. Geographical data capture is less and less expensive because of the evolution in tools and techniques: GPS, digital images with high resolution, automated correlation in photogrammetry, automated image analysis in remote sensing, and so on. But the diversity of geographical data does not appear simply because of technical ability; it also originates in a deep-rooted need of geographers to simultaneously manipulate several points of view in order to analyse the geographical world. The best example of this is the diversity of existing maps with different scales, but also different contents and purposes, from topographic to thematic maps.

We thus face a strong diversity of geographical information and consequently differences between geographical databases describing one same area. These differences can have several origins. First of all, databases (DB) have different purposes and thus different contents, organizations, and granulometry. For instance, a spatial DB for urban management does not include the same geographical entities as a spatial DB intended for applications in landscape ecology; in the same way, representation of objects in a map to a 1:25 000 scale differs from that associated with a map to a 1:250 000 scale. Second, data are produced from different and various sources: what is captured from an old map is different from what is captured from spatial images or field surveys. Third, the sources, even if they are identical, may be differently interpreted and digitized because of the complexity of the geographic data capture: it requires identifying, selecting, splitting, merging, and delineating geographical phenomena (Gesbert 2004, Uitermark *et al.* 2005). Fourth, the geographical world is in constant evolution, whereas data give only the situation at certain dates. And one notices that even one database usually originates from various sources with different dates and is thus heterogeneous in this respect. Finally, the data-capture process is still complex, and databases also contain errors leading to differences between databases. All these reasons lead to differences between data representing the same geographic entities in various respects (Parent *et al.* 1996).

Most of the time, the differences reflect different points of view, and are thus of interest. But, in order to make the best of these differences, a key issue is to determine if the differences are somehow normal, or if they are due to erroneous or outdated data in one database (Egenhofer *et al.* 1994, Sheeren *et al.* 2004b). For example, typical differences and inconsistencies are shown in figure 1. Figure 1(*a*) shows important differences in the way of representing crossroads, but they are only due to the differences between levels of details. According to the class definitions of the two DB and the capture rules of the objects, the differences are normal. Figure 1(*b*) shows other differences in the way of representing crossroads, but in this

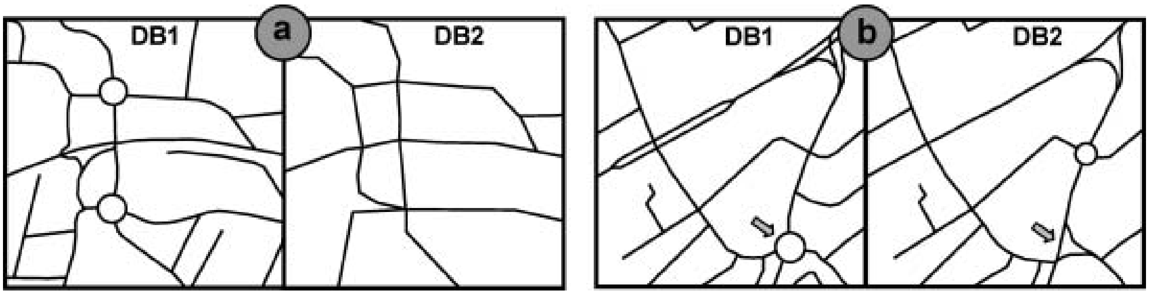


Figure 1. Typical differences between two topographic databases (consistent (*a*) and inconsistent (*b*) representations).

case there surely is an inconsistency at the crossroad pointed out: it is very unlikely that the same crossroads should be represented by a roundabout in one database and by Y-shaped crossroads in the other database. In this case, one can easily imagine that one DB is more up to date than the other one, but we do not know which one without additional information.

In order to efficiently combine different views of the world, one must identify inconsistencies like that shown in figure 1. Identifying inconsistencies may be useful either to correct them when possible, or to point out potential problematic areas that need surveying, or even to characterize the degree of certainty of the information (Goodchild and Jeansoulin 1998). While, in the past decades, the main issue was the acquisition of geographic data, this new era has to deal with combining all this information. Managing consistency has therefore become a key issue. This is perfectly illustrated in Europe by the INSPIRE directive that states that 'the implementing rules shall be designed to ensure consistency between items of information which refer to the same location or between items of information which refer to the same object represented at different scales' (Directive of the European Parliament and of the Council establishing an Infrastructure for Spatial Information in the European Community (INSPIRE), Joint text approved by Conciliation Committee, 17/01/2007). In this context, this article proposes an approach to identify inconsistencies between geographical data. After this introduction, section 2 will refine the definition of the issue: we will describe how this task is carried out within the wider field of geographic database integration, and we will define precisely what we consider to be an inconsistency. In section 3, we will show that the inconsistency detection can be thought of as a knowledge-acquisition problem, and section 4 will introduce which sources of knowledge can be used to perform the knowledge acquisition, namely textual specifications and data themselves. Section 5 will then propose two approaches to actually perform the knowledge acquisition from data with data-mining techniques. Section 6 will present experiments illustrating the suggested approaches on three test cases. Finally, before concluding, in section 7 we will discuss and compare our approaches.

2. Detecting inconsistencies in the framework of spatial database integration

2.1 *Spatial database integration*

For quite some time, integration has undergone substantial research within the database community (Batini *et al.* 1986, Parent and Spaccapietra 2000). There currently exist several paradigms to provide a unified and coherent view of data stored in multiple sources: multi-database systems (Litwin *et al.* 1990), federated systems (Sheth and Larson 1990), mediated systems (Wiederhold 1992) or datawarehousing (Calvanese *et al.* 2001). This integration issue has also been addressed in the field of spatial databases, and is considered as an important issue for data producers who need to create and maintain various databases, but also for data users who require up-to-date and rich data (Sester *et al.* 1998, Kilpeläinen 2000, Hampe and Sester 2002, Friis-Christensen 2003, Sheeren *et al.* 2004a, Mustière and van Smaalen 2007). Explicitly defining the relationships between heterogeneous data sources can help keep the databases up to date by propagating updates from the more detailed to the less detailed (Lemarié and Badard 2001). It also increases the potentiality of applications which can benefit from using databases with multiple representations.

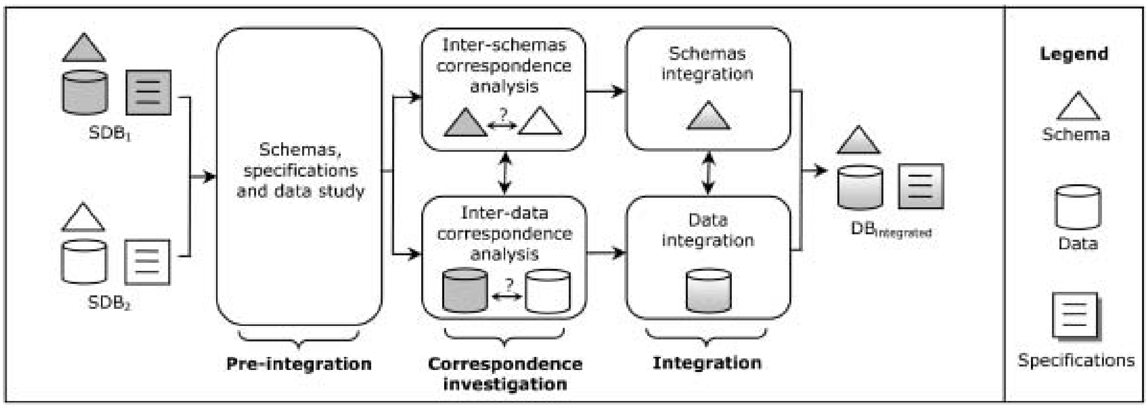

SDB1
SDB2
Schemas, specifications and data study
Inter-schemas correspondence analysis
?
Inter-data correspondence analysis
?
Schemas integration
Data integration
DBIntegrated
Pre-integration
Correspondence investigation
Integration
Legend
Schema
Data
Specifications

Figure 2. General framework of spatial databases integration (Devogele *et al.* 1998, Sheeren *et al.* 2004a).

A general framework for database integration has been adapted to geographical databases (Devogele *et al.* 1998, Sheeren *et al.* 2004a), leading to figure 2.

Pre-integration consists in the study of each database, to gain a good understanding of its content, and to prepare the integration. *Correspondences investigation* aims at identifying and declaring correspondences between the elements of the schemata and between objects of the databases. Even if the schema and object levels are clearly dependent, geographic databases specifically require separation of these investigations: because of the lack of universal identifiers on objects, correspondences at the objects level are not directly derived from correspondences at the schema level. Finally, the *Integration* step is the actual filling of the integrated database. A new schema is defined, and, according to the integration strategy adopted (i.e. mono-representation or multi-representation), objects are merged or linked and transferred into the new system. The mono-representation strategy provides a unique representation of the world which relies on the merging of the more detailed data from the initial DBs. In this case, no link between the new system and the initial DBs is retained. The initial data cannot be inferred from the integrated DB. On the contrary, in the multi-representation strategy, the respective representations are preserved, and explicit relationships between the homologous objects are created. The integrated database, which can take the form of a federated system, remains compatible with the initial DBs and includes the same geographical entities represented with different levels of details. This framework allows us to provide a brief overview of some of the contributions on database integration and their link with the issue of identifying inconsistencies between datasets.

At the schema level, methodologies have been defined in order to produce a single unified description of the originally independent schemata and resolve conflicts between concepts (Branki and Defude 1998, Devogele *et al.* 1998, Strauch *et al.* 1998, Park 2001, Balley *et al.* 2004). The current trend relies on solutions based on mediation or the use of ontologies (Leclercq *et al.* 1999, Visser *et al.* 2002, Fonseca *et al.* 2003, Gesbert 2004, Rodriguez and Egenhofer 2004, Uitermark *et al.* 2005). New models supporting multiple representations have also been put forward (Vangenot *et al.* 2002, Friis-Christensen 2003, Mustière and van Smaalen 2007). Some other works establish correspondences at the schema level from correspondences observed at the data level (Duckham and Worboys 2005, Volz 2005). This latter approach particularly requires identification of inconsistencies at the data level, in order to avoid propagating them erroneously to the schema level.

At the data level, a typology of differences encountered between databases has been established (Parent *et al.* 1996). Geometric feature matching algorithms have been developed to establish explicit links between objects in different representations. Some are dedicated to the detection of updates (Lemarié and Badard 2001, Gomboši *et al.* 2003), while others compare databases with similar levels of details but different purposes (Sester *et al.* 1998, Walter and Fritsch 1999, Beeri *et al.* 2004, Volz 2006), and some concentrate on databases with different levels of details (Devogele 1997, Haunert 2005, Mustière 2006). In those works, the issue of consistency assessment appears in several ways. Studies on data matching, when applied to real data, observe inconsistencies between the data. When they matched the German ATKIS and GDF databases, Walter and Fritsch (1999) observed that 'problems can occur in areas with completely different acquisition where there is no reasonable matching possible'. In several previous experiments, we also observed inconsistencies between the French BDTOPO and BDCARTO databases: some of the inconsistencies were identified once the matching was carried out, for example when automatically comparing attributes values of matched data; other inconsistencies have been identified because they resulted in abnormal unmatched data (Mustière 2006).

More globally, works comparing maps or databases are faced with the issue of explicitly managing fuzziness or uncertainty, and this is partly due to the presence of inconsistencies between the data (Worboys and Clementini 2001, Ahlqvist *et al.* 2003, Comber *et al.* 2004, Fritz and See 2005, Hagen-Zanker *et al.* 2005). As noticed by Duckham and Worboys (2005): 'Addressing the problem of degenerate fusion products is important to the success of automated information fusion. The problem will be a key area of future research [...]'.

As far as we know, only a few studies have specifically addressed the issue of consistency assessment between multiple representations in spatial databases, and in general, they focused on the consistency of topological relations (Egenhofer *et al.* 1994, El-Geresy and Abdelmoty 1998, Paiva 1998). Those works define a formal model to qualitatively describe topological relationships in spatial scenes, and thus compare qualitative descriptions. In a general way, these models are useful to express rules formally and unambiguously such as: 'if the relation is *Disjoint* in DB1 it should not *Overlap* in DB2' (a detailed description of these works can be found in Rodriguez 2005). However, the issue of acquiring the knowledge, which is necessary to build these rules, is rarely explicitly addressed.

In this context, our work aims at defining a global framework for assessing consistency between databases, and particularly to study which source of knowledge can be used for that. In the general framework of spatial database integration presented previously (figure 2), this work is part of the *Correspondence investigation* step. More precisely, our work concerns the evaluation of correspondences discovered once instance-level elements have been matched. We strongly believe that if matching data is a prerequisite for integration, it is not the only one: matched data must be analysed and inconsistencies discovered, in order to be efficiently handled when actually integrating data.

2.2 *Definition of inconsistency*

Before assessing the consistency between multiple representations, one must define what an inconsistency is. Even if its meaning seems rather naturally understood, we believe that this concept needs clarifying. Egenhofer *et al.* (1994) define consistency as follows:

> *Consistency* refers to the lack of any logical contradiction within a model of reality. This must not be confused with *correctness*, which excludes any contradiction with reality. […] In itself, each individual level may be consistent, however, when integrating and comparing the different levels, inconsistencies may be detected if the representations contradict.

This clear definition allows us to avoid any misunderstanding. First of all, it should be clear that what is addressed in this paper is the consistency *between* representations. This should not be confused with (internal) consistency, which is the respect of a model. For example, an internal consistency constraint within one database may be that 'for all objects of a class, values of a given attribute must be between some given thresholds'. This is beyond the scope of this paper. Second, consistency refers to the absence of contradictions between representations, but not to the correctness of the data. Determining the correctness of the data is a matter of quality control, and requires an external reference source like a field survey. In our context, we do not use any reference that is known to be error-prone; nor do we consider that one database is more correct than another, but we search for contradicting representations of the same geographic phenomenon in two different databases. This consequently may lead to the detection of incorrect data in one of the databases, because if the data were correct, there would not be the least inconsistency between them. But in our case, we usually ignore which database is incorrect.

However, Egenhofer *et al.*'s definition still needs clarifying in order to specify what 'contradict' means. As shown in figure 1, determining inconsistencies may require some complex knowledge defining when the representations are in contradiction. This knowledge is actually closely related to the elements that best define the database: its specifications. Indeed, a database is never captured by chance: the data-capture process follows some rules expressed in the specifications of the database (figure 3). These specifications can be more or less explicit and precise, but they always exist, even if in the worse case it is only informally in the mind of the human operator filling the database. Based on this observation, we propose the following definition of inconsistency:

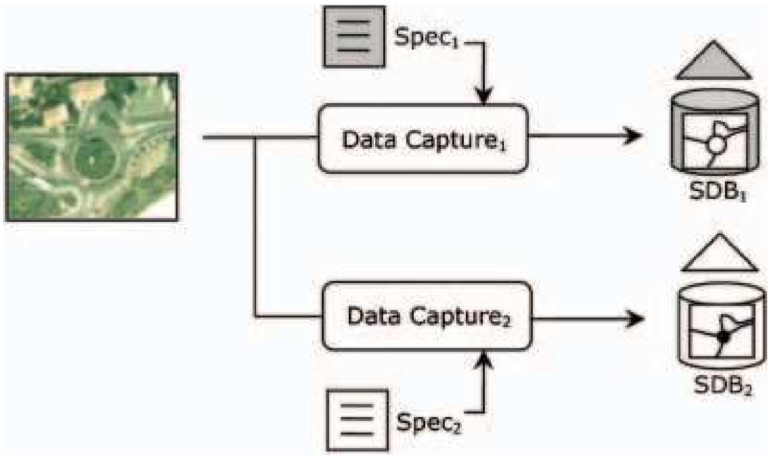


Figure 3. Specifications govern the representation of geographical phenomena in databases.

> Definition (inconsistency): 'Two representations of a given geographic phenomenon are said to be inconsistent if and only if the differences between these representations cannot be explained by their respective database specifications. Otherwise, the representations are said to be consistent'.

It should be noted that in this definition, the notion of difference must be taken in the wide sense of 'the result of the comparison', including possibly the result 'exactly identical'. Actually, even if a phenomenon is represented exactly in the same way in two different databases, the two representations may be inconsistent if the specifications express that these objects should not be identical. For example, if one database specifies that buildings should be represented by a polygon delimitating the roof, and another database specifies that the polygon should be located at ground level, and if a given building has exactly the same shape and height in the two databases, this is inconsistent.

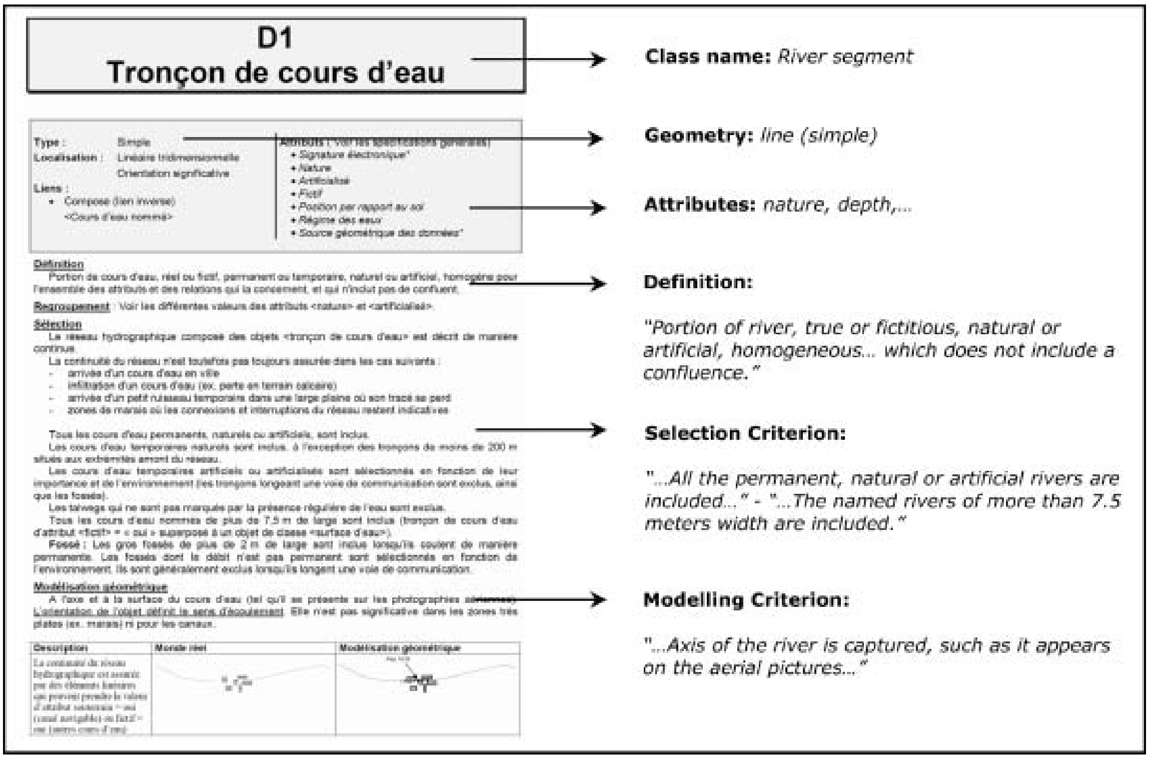
3. Sources of knowledge used for assessing consistency: specifications and data

As mentioned before, determining inconsistencies may require some complex knowledge defining when the representations contradict each other. In this section, we will present two different sources for knowledge: textual specifications and DB instances themselves.

3.1 *Specifications*

According to our definition, inconsistencies are differences that cannot be explained by the specifications. The first idea to acquire the necessary knowledge to assess consistency is thus to make use of the specifications themselves. Most data producers have such specifications in the form of textual documents. These are necessary at least to guide data capture and to ensure a certain homogeneity in the data if several people are involved in the data-capture process, which is almost always the case for geographic data that are voluminous and take a long time to acquire. The specifications may also be provided to users as they precisely describe the semantics of the data. They are an integral part of metadata. For spatial database integration, the specifications seem to be a fundamental source of knowledge. Even if some information about the precise semantic of data is already captured in the database schema, it only concerns a small part of it and is therefore not sufficient. For instance, no information about the spatial data-capture constraints is given in the schemata, even in conceptual spatial-temporal models (Vangenot *et al.* 2002). Consequently, while the correct interpretation of the schemata mainly relies on the database administrator's expertise and data dictionary for traditional databases integration (Parent and Spaccapietra 2000), for data semantic interpretation the specifications for spatial databases need to be used.

Figure 4 shows an extract of the specifications of the French IGN BDTOPO® database (specifications are in French; the full text is long and not legible on the figure, but typical excerpts are translated in English). For each class of the database, these specifications describe which real-world entities are represented in this class and how they are represented. For instance, they answer the typical questions 'What does *river* mean in the database?', 'Is an aqueduct a *river* in the database sense?', 'Which rivers may be represented in the class *river?*', and 'How is a *river* represented in the database?'.



D1
Tronçon de cours d'eau

Type : Simple
Localisation : Linéaire tridimensionnelle
Orientation significative
Liens :
- Composé (lien inverse) <Cours d'eau nommé>

Attributs (voir les spécifications générales)
- Signature électronique*
- *Nature*
- *Artificialisé*
- *Fictif*
- *Position par rapport au sol*
- *Régime des eaux*
- *Source géométrique des données**

Définition
Portion de cours d'eau, réel ou fictif, permanent ou temporaire, naturel ou artificiel, homogène pour l'ensemble des attributs et des relations qui la concernent, et qui n'inclut pas de confluent.

Regroupement : Voir les différentes valeurs des attributs <nature> et <artificialisé>.

Sélection
Le réseau hydrographique composé des objets <tronçon de cours d'eau> est décrit de manière continue.
La continuité du réseau n'est toutefois pas toujours assurée dans les cas suivants :
- arrivée d'un cours d'eau en ville
- infiltration d'un cours d'eau (ex. perte en terrain calcaire)
- arrivée d'un petit ruisseau temporaire dans une large plaine où son tracé se perd
- zones de marais où les connexions et interruptions du réseau restent indicatives

Tous les cours d'eau permanents, naturels ou artificiels, sont inclus.
Les cours d'eau temporaires naturels sont inclus, à l'exception des tronçons de moins de 200 m situés aux extrémités amont du réseau.
Les cours d'eau temporaires artificiels ou artificialisés sont sélectionnés en fonction de leur importance et de l'environnement (les tronçons longeant une voie de communication sont exclus, ainsi que les fossés).
Les talwegs qui ne sont pas marqués par la présence régulière de l'eau sont exclus.
Tous les cours d'eau nommés de plus de 7.5 m de large sont inclus (tronçon de cours d'eau d'attribut <fictif> = « oui » superposé à un objet de classe <surface d'eau>).
Fossé : Les gros fossés de plus de 2 m de large sont inclus lorsqu'ils coulent de manière permanente. Les fossés dont le débit n'est pas permanent sont sélectionnés en fonction de l'environnement. Ils sont généralement exclus lorsqu'ils longent une voie de communication.

Modélisation géométrique
A l'axe et à la surface du cours d'eau (tel qu'il se présente sur les photographies aériennes). L'orientation de l'objet définit le sens d'écoulement. Elle n'est pas significative dans les zones très plates (ex. marais) ni pour les canaux.

Description	Monde réel	Modélisation géométrique
La continuité du réseau hydrographique est assurée par des éléments linéaires qui peuvent prendre la valeur d'attribut [illegible] = oui (canal navigable) ou fictif = oui (autres cours d'eau)		

Class name: *River segment*

Geometry: *line (simple)*

Attributes: *nature, depth,…*

Definition:

"Portion of river, true or fictitious, natural or artificial, homogeneous… which does not include a confluence."

Selection Criterion:

"…All the permanent, natural or artificial rivers are included…" - "…The named rivers of more than 7.5 meters width are included."

Modelling Criterion:

"…Axis of the river is captured, such as it appears on the aerial pictures…"

Figure 4. Elements encountered in the textual specifications (translated from IGN BD TOPO® database).

From these individual descriptions of the semantics of the schema of each database, one may derive all the information necessary to compare them and assess their consistency. Questions such as the following may be answered by comparing the specifications: 'If an object in database SDB_1 belongs to class *river*, in which class(es) may it have a corresponding object in SDB_2?', 'Does any object of class *river* in SDB_1 have a corresponding object in class *watercourse* of SDB_2?', and 'If a class *river* object in SDB_1 has certain properties, how should we expect the possible corresponding object(s) in SDB_2 to be represented?'.

Theoretically speaking, all the necessary information is then explained in the specifications, and hardly anywhere else (Gesbert 2004). But exploiting them practically is an arduous task. First of all, the description of the capture constraints in a natural language is mostly informal. The specifications are also voluminous, and useful information for a specific problem may be split into several pieces that may be hard to discover among the huge quantity of information available. The specifications may also be organized using different structures for different databases. For example, one may have a 'selection criteria' section while this information may be found for other databases either in a 'definition' section, or even in the definitions of the possible values of the attributes. In addition, specifications frequently require an interpretation. For instance, we can find descriptions like: 'rivers are captured as regards to their importance and their surroundings' or 'only the main objects are captured'. These natural language descriptions suffice for the people in charge of the production of the SDB because they use their knowledge and their know-how. However, in an automation context, this knowledge is not always sufficient and not well adapted. The formalization of the specifications could be a first step toward automatic analysis (Mustière *et al.* 2003, Gesbert 2004). We think that capture constraints should benefit considerably from being transformed into a more formal or even computational language. This is the way to introduce more tractable semantics in the metadata. However, as this task is arduous, we propose another source of knowledge to acquire specifications: the data itself.

3.2 *Data*

Comparing data in order to compare schemata has already been proposed in recent years (Duckham and Worboys 2005, Volz 2005, Tomai 2006). The idea could be extended to the consistency assessment as illustrated in figure 5. This figure shows the superimposition of two classes from two different databases: grey surfaces are *water surfaces* in the French IGN BDTOPO® database (roughly a database with a metric precision), and black points are *punctual water names* in the French IGN BDCARTO® database (roughly a database with a decametric precision). Just looking at the data, without studying the textual specifications, one may extract a considerable amount of knowledge from this image. It seems there is a point for each surface that is (1) not part of a river and (2) large enough. The threshold defining the 'big enough' can even be evaluated: it is between the area of the smallest surface with a corresponding point and the area of the largest surface without any corresponding point. Additionally, one may think that adjacent surfaces may be merged before measuring the area (see the large surface split on the left of the image). Examination at the attributes of the data (not displayed here) would even provide us with more information: we may see that the attribute 'toponym' of one database has values close to that of the attribute 'name' for corresponding objects in the other. Based on this knowledge extracted from a simple excerpt of the data, one may consider that if we encounter somewhere else a very small surface with a homologous point in the other database, this may certainly be an inconsistency.

Analysing data in order to extract knowledge, or in other words 'data mining', thus seems to be useful for consistency assessment. To take advantage of this idea, the key issue is then to define a methodology to actually perform the knowledge acquisition. This will be detailed hereafter.

4. Knowledge-based approach for assessing consistency

In this section, we will present an approach for assessing consistency between multiple representations that derives in a natural way from the considerations of the

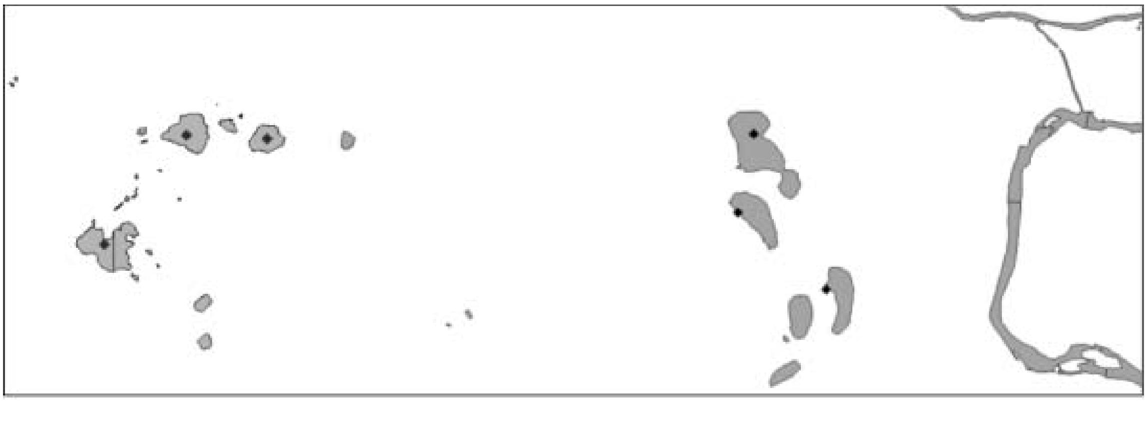


Figure 5. Comparing data in order to extract rules for assessing consistency (excerpts of water surfaces of the IGN BDTOPO® and punctual water of the IGN BDCARTO® databases).

previous sections (Sheeren *et al.* 2004b, Sheeren 2005). The proposed approach is summarized in figure 6. Data are input of our process. The output of the process is the set of consistent and inconsistent pairs detected and justified. The consistency assessment is realized by applying the MECO method (Method for Evaluating the COnsistency). This method is guided by some domain knowledge, which is obtained by applying the MACO method (Method for Acquiring knowledge to evaluate the COnsistency). MACO extracts the useful knowledge from data. The knowledge is formalized in a knowledge base and appears at the interface of the two complementary methods.

MECO is applied on each set of homologous objects of the two DB that correspond to the same geographical entities. To carry out MECO and according to the integration process presented in figure 2, we make the assumption that correspondences at the schema level have already been declared in terms of Interdatabases Correspondence Assertions (Devogele *et al.* 1998, Parent and Spaccapietra 2000, Sheeren *et al.* 2004a). The first two main parts of the approach, MECO and the knowledge base that guides it, are discussed in this section. The last part, MACO, which represents the key point of the approach presented here, will be described in section 5.

4.1 *MECO: Method for evaluating consistency*

Differences between representations in the spatial databases that need to be integrated are detected and analysed in MECO. This problem-solving method consists in several steps, each associated with specific tools. MECO is fully automated. Its core is detailed in figure 7.

MECO's first step is *enrichment*. Its goal is to reduce heterogeneity between data from the initial two databases. It facilitates the comparison between databases and prepares data for checking compliance with some internal integrity constraints. The enrichment phase mainly consists in extracting implicit spatial concepts from data that exist only through geometry and to which the specifications explicitly refer (i.e. qualifying the spatial properties of the objects like shape and dimension, or extracting spatial relations and implicit objects). The essential properties of spatial

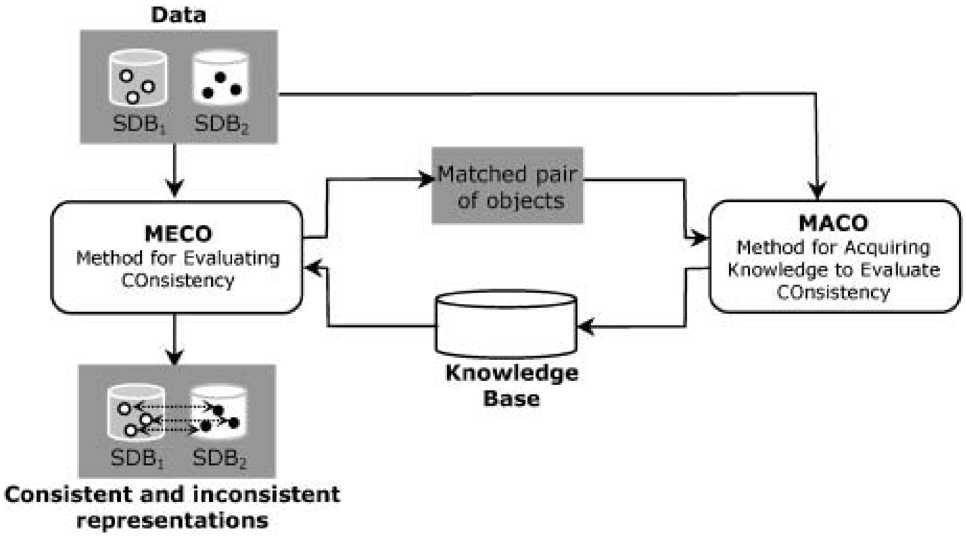

Data
SDB1
SDB2
Matched pair of objects
MECO
Method for Evaluating COnsistency
MACO
Method for Acquiring Knowledge to Evaluate COnsistency
Knowledge Base
SDB1
SDB2
Consistent and inconsistent representations

Figure 6. Overall knowledge-based approach for assessing consistency between multiple representations.

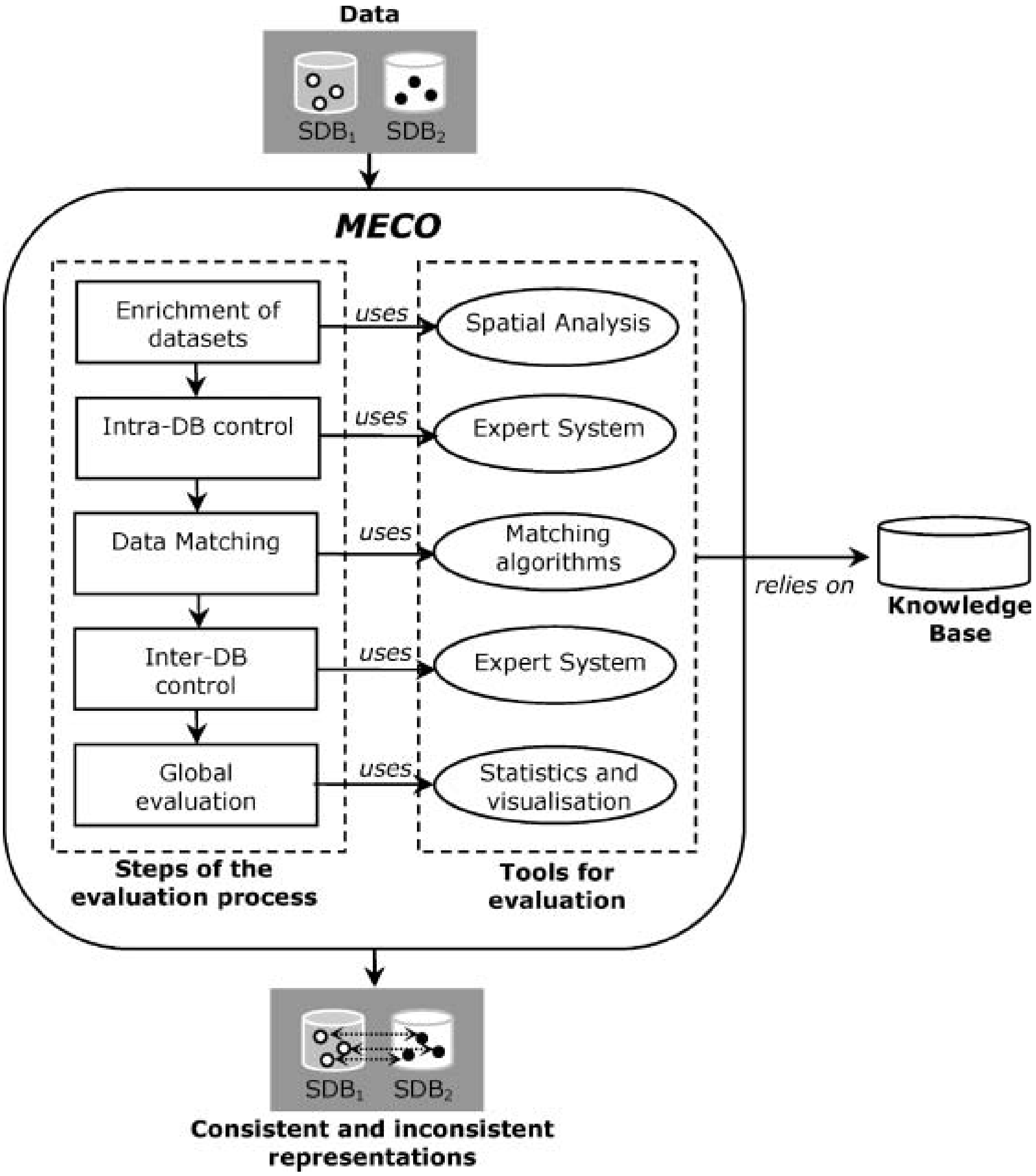


Figure 7. MECO: steps and tools to evaluate the consistency.

concepts which are seldom stored as database attributes are thus made explicit. All of this results in the creation of new classes and attributes, or in the reorganization of existing ones. An example of such enrichment would be the merger of adjacent lake parts in order to create actual lakes (cf. section 3.2). The enrichment is a traditional task in the databases integration methodologies (Parent and Spaccapietra 2000). Its utility depends on the richness and the heterogeneity of the initial database schemata. For spatial databases, the enrichment requires specific spatial analysis measures and geometrical algorithms to be called upon.

The second step of MECO is the *intra-database control*. This step is applied to both databases individually and may be viewed as spatial-data integrity constraint checking (Cockroft 1997, Servigne *et al.* 2000). Part of the specification is checked so as to detect any internal errors and to determine how the data instances globally respect the specifications. In this step, only the specifications that can be evaluated within one database without external data are considered. The checking is related to several spatial properties and attributes compared with their definition domain (e.g.

the minimal size that an object must respect). Thus, only logical consistency errors are pointed out. Classification confusions or positional errors in particular are not detected at this step. The intra-database control is performed automatically by an expert system, relying on rules determined by the MACO method (section 5).

Once internal constraints have been checked, databases may be compared. So, the next step is the spatial *data matching* (also called *conflation*). Correspondences between the data are computed, and pairs of homologous objects are created. Whenever possible, the matching tools used in this approach are mainly based on the comparison of the position of the objects, in addition to their geometrical and topological properties. Such tools are relatively independent of the data and exploit only few heuristics. At this level, we are not concerned yet with the conformity of representations with their specifications. The homologous objects that do not strictly respect their specifications can be matched. This enables us to consider the matching process as a 'black box' guided by general knowledge about spatial data but with as little knowledge as possible specific to the compared databases.

The next step is the key step: the actual *inter-DB control* which follows the data-matching step. It consists in comparing representations of the matched objects and checking consistency. Each representation is analysed in a cross way by taking into account the representation of the homologous object. The consistency assessment is based on the use of some domain knowledge specific to the compared database. All the spatial and non-spatial properties which require another source of data to be analysed are now considered. The inter-DB control is thus complementary to the intra-DB control, and its results are exploited to explain and make assumptions on the origin of the inconsistencies detected. For instance, if an inconsistency between two representations is detected at the inter-DB control and if one of the two representations corresponds to an internal error according to the intra-DB control, we can deduce that the inconsistency certainly arises from an internal error of this DB. At the end of this step, the matched pairs are qualified as consistent or inconsistent. The inter-DB control is performed by means of an expert system that relies on rules determined by the MACO method. The creation of these rules is the main subject of our approach.

A *global evaluation* is supplied at the end of MECO to combine and summarize the results. It gives the number of consistent and inconsistent representations, their type, and how significant they are.

4.2 *Knowledge representation and management*

In knowledge-based approaches, one important aspect is how knowledge is represented in an appropriate computer-usable form. Many theoretical frameworks, often related to the Artificial Intelligence (AI) field, exist for knowledge representation and management. In our approach, the knowledge used in the intra- and inter-database controls is encoded in the form of *if–then* rules. These rules are managed automatically by an expert system. For the inter-database control that is detailed more in depth in this paper, since it is the core of our approach, we propose in particular to organize the knowledge in the rule base in two different ways: following either the *direct classification* approach or the *predictive* approach, as described hereafter.

4.2.1 Direct classification approach. In this first approach, the knowledge is represented in the form of a set of rules that directly explain each difference between representations. More precisely, if (o_{1i},o_{2j}) represents a matched pair of objects, the direct classification of differences consists in the activation of rules such as:

if condition$_A$ (o_{1i}, o_{2j}) then the pair (o_{1i}, o_{2j}) is consistent.
if condition$_B$ (o_{1i}, o_{2j}) then the pair (o_{1i}, o_{2j}) is inconsistent.

For objects coming from two databases with different resolutions, an example of such rules could be:

if the pair (o_{1i}, o_{2j}) is composed of a house o_{1i} represented by a polygon smaller than 200 m^2 and a building o_{2j} represented by a point then the pair (o_{1i}, o_{2j}) is consistent.

In practice, the rules related to inconsistent representations may not be formulated. Only the rules enabling consistent representations to be detected may be defined. The number of these rules is limited, and inconsistencies can be deduced from them. The correspondences that do not respect these rules (i.e. for which no rule was activated by the inference engine of the expert system) are considered as inconsistent.

4.2.2 Predictive approach. In the predictive approach, the matching pairs are not labelled directly as inconsistent or consistent. The consistency is assessed in several steps. First, each representation of a matching pair is used to predict the conditions the representation of the homologous object has to respect in the other database. So, we determine for instance the conditions relating to the shape of the objects in SDB$_1$ using those in SDB$_2$, and the conditions relating to the shape of the objects in SDB$_2$ using those in SDB$_1$. Then, the predicted conditions on the representations are compared with the actual representations stored in the SDBs. If the actual representations respect the conditions in both directions, the pair of homologous objects can be considered as consistent; otherwise the pair of homologous objects is labelled as inconsistent. In terms of rules, this approach can be expressed as follows:

if condition$_A$ (o_{1i}) then the object (o_{2j}) must respect condition$_B$
if condition$_C$ (o_{2j}) then the object (o_{1i}) must respect condition$_D$

Knowing the rules that need to be respected (let us say here condition$_B$ and condition$_D$), we can easily test the consistency by means of the following rule:

if condition$_B$ (o_{2j}) and condition$_D$ (o_{1i}) then the pair (o_{1i},o_{2j}) is consistent else it is inconsistent

The rules expressed in such a way are easy for a human being to understand, but they can also be directly transformed in a more syntactically standard form for an expert system as:

if condition$_A$ (o_{1i}) and condition$_B$ (o_{2j}) then the pair (o_{1i}, o_{2j}) is consistent else it is inconsistent.
if condition$_C$ (o_{2j}) and condition$_D$ (o_{1i}) then the pair (o_{1i}, o_{2j}) is consistent else it is inconsistent.

Figure 8 is an example to illustrate this approach. In SDB$_1$, a particular building is shown with a detailed representation (the contour of the object). Its area is 13 square metres. In SDB$_2$, the object is represented with a node. Let us imagine that the specifications in SDB$_1$ indicate 'the object must have a detailed representation if and only if the area is larger than 10 square metres' and those in SDB$_2$ say 'the object

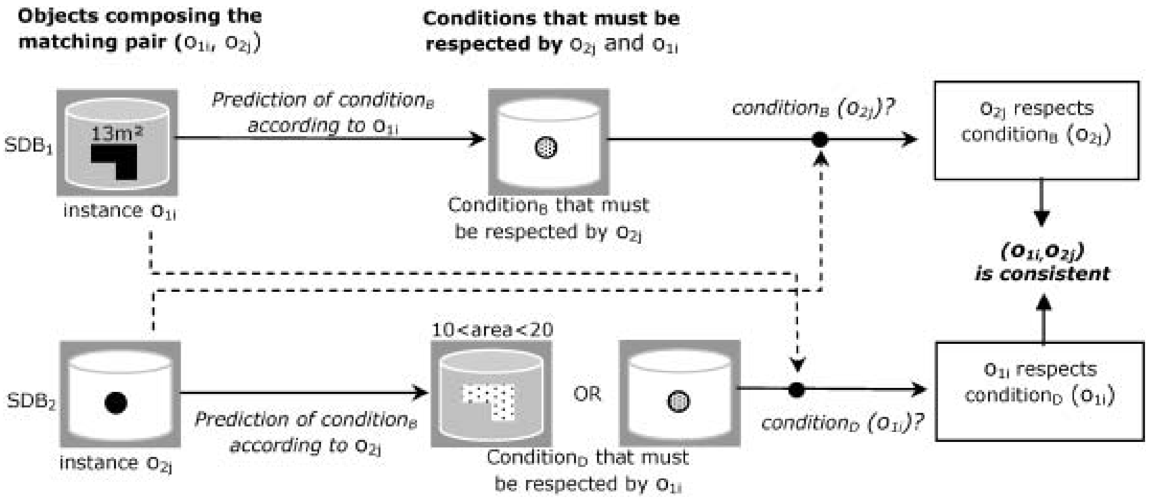

Objects composing the matching pair (o_{1i}, o_{2j})
Conditions that must be respected by o_{2j} and o_{1i}
SDB$_1$
13m²
instance o_{1i}
Prediction of condition$_B$ according to o_{1i}
Condition$_B$ that must be respected by o_{2j}
condition$_B$ (o_{2j})?
o_{2j} respects condition$_B$ (o_{2j})
(o_{1i},o_{2j}) is consistent
SDB$_2$
instance o_{2j}
Prediction of condition$_B$ according to o_{2j}
10<area<20
OR
Condition$_D$ that must be respected by o_{1i}
condition$_D$ (o_{1i})?
o_{1i} respects condition$_D$ (o_{1i})

Figure 8. Implementing the inter-databases control following the predictive approach.

must have a detailed representation if and only if the area is larger than 20 square metres'. This leads to the following rule: the object in SDB$_2$ should be represented with a node (condition$_B$) if the object in SDB$_1$ is a point or detailed but less than 20 m in size (condition$_A$). On the other hand, it also gives another rule: the object in SDB$_1$ should be represented with either a node or a detailed representation with an extent ranging between 10 and 20 square metres (condition$_D$) if the object in SDB2 is represented with a node (condition$_C$). As, in our example, objects in SDB$_1$ and SDB$_2$, respectively, meet condition$_A$ and condition$_C$ defined above, we can thus predict that the objects in SDB$_1$ and SDB$_2$ should respectively meet condition$_D$ and condition$_B$. Having determined these conditions, we can compare them with the actual representations stored in the SDB. In this particular case, we can conclude that the representations are consistent, since the conditions are respected.

5. MACO: method for acquiring knowledge to evaluate consistency

Expert systems have already proved their efficiency in many applications when knowledge needs to be introduced (David *et al.* 1993, Leung and Leung 1993, Bonnett *et al.* 2004). However, the key issue in their implementation is knowledge acquisition. It is often difficult to grasp knowledge directly from human domain experts. They are rarely able to supply an explicit description of the knowledge they use for a given problem: 'as the expert acquires expertise, his declarative knowledge [of which he is aware] becomes procedural and he loses conscience of what he knows' (Musen 1993). This problem is known as the *knowledge acquisition bottleneck* (Feigenbaum 1981) and has also been explicitly pointed out in the spatial domain (Weibel *et al.* 1995, Mustière *et al.* 2000, Sester 2000, Mustière 2005).

In our approach, knowledge acquisition is also an important issue. As explained in section 3, the knowledge we use to assess consistency may originate from two sources: the specifications and the data. However, as we already observed, extracting knowledge automatically from the specifications is a difficult task, and encoding it manually in a computational language is probably too long in practice. So, the main source of knowledge we exploit here is data, using data-mining techniques (Witten and Frank 2005) and in particular supervised machine learning (Mitchell 1997). These methods originated in the AI field, although they are related to the inferential statistics.

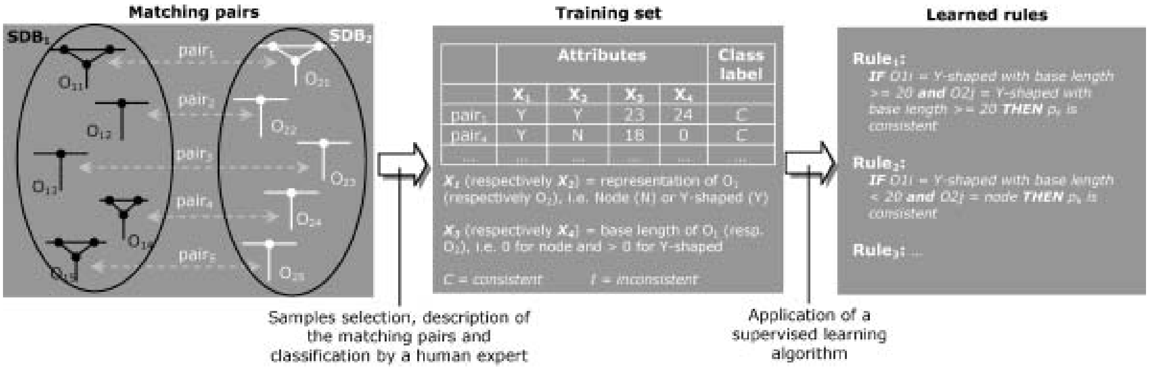
Supervised learning requires that a set of training examples of a concept be given by a 'teacher' to automatically induce a general model that will capture the general patterns in the training data (Mitchell 1997). An expert gives some examples in the form of, on the one hand, a description of an object and, on the other hand, a classification (label) of this object. Learning algorithms automatically build a model for these examples so as to best predict their class. This model can then be applied to predict future, previously unseen observations with the highest accuracy today's algorithms may afford.

There are a wide variety of algorithms from numerical algorithms such as Artificial Neural Network (ANN), Support Vector Machines (SVM) as well as more symbolic algorithms such as decision trees or decision rules. One very popular supervised machine-learning algorithm is the C.4.5 classifier proposed by Quinlan (1993). This algorithm enables the creation of decision trees that predict class membership by recursively partitioning a dataset into more homogeneous subsets using discriminating attributes. The decision tree can subsequently be transformed into if–then rules. The algorithm requires that the training examples be represented as a list of attributes (i.e. expressed in the 'attribute-value' representation language). The attributes are selected to generate the decision nodes of the tree by computing the *information gain ratio*. This measure evaluates the reduction in entropy in the data produced by a split. The classification of the data that maximizes this reduction is retained. Results obtained by C4.5 are easy to interpret and validate. Their simplicity helps to explain why observations are classified or predicted in a particular manner. The rules can then be revised by a domain expert, should this be necessary. This symbolic algorithm (in opposition to numerical approaches) is particularly well adapted for building our knowledge base. We have used it for our experiments, as it has been proven experimentally that it gives the best predictive accuracy performances on numerous applications. The theoretical explanation of this behaviour lies in its information gain ratio measure.

5.1 *Learning direct classification rules from data*

The direct classification approach is characterized by the definition of a set of rules that enable one to determine directly if a matching pair is consistent or not (see section 4.2.1). Implementing this approach with supervised learning is relatively simple. A training example corresponds to a characterized matching pair of objects, i.e. a pair represented by a vector of descriptive attributes. These attributes translate the initial representation of each object constituting the pair in a symbolic way. The training example is also labelled. The label indicates if the matching pair is consistent or inconsistent. From a set of such training examples describing the differences, direct classification rules can be induced from data. This process is illustrated schematically in figure9. The example shows differences between representations of triangle junctions (i.e. Y-shaped crossroads). Notice that the intervention of a human domain expert (the supervisor) is required in the process. The expert has to determine the label for each of the collected examples, validate the learned rules, and evaluate their relevance (which is the case for any learning method).

In this approach, an interesting aspect is that the knowledge acquired can be slightly different from the knowledge existing in textual specifications. Since training examples are classified by the expert, it is the expert's point of view which is learned, with the implicit rules that they use to determine if the differences are consistent or



	Attributes				Class label
	X_1	X_2	X_3	X_4	
$pair_1$	Y	Y	23	24	C
$pair_4$	Y	N	18	0	C
...	...	...	...	...	...

X_1 (respectively X_2) = representation of O_1 (respectively O_2), i.e. Node (N) or Y-shaped (Y)

X_3 (respectively X_4) = base length of O_1 (resp. O_2), i.e. 0 for node and > 0 for Y-shaped

C = consistent *I = inconsistent*

$Rule_1$:
IF *O1i = Y-shaped with base length >= 20* ***and*** *O2j = Y-shaped with base length >= 20* ***THEN*** *p_i is consistent*

$Rule_2$:
IF *O1i = Y-shaped with base length < 20* ***and*** *O2j = node* ***THEN*** *p_k is consistent*

$Rule_3$: ...

Figure 9. Learning process for acquiring direct classification rules from matching pairs reformulated as training examples.

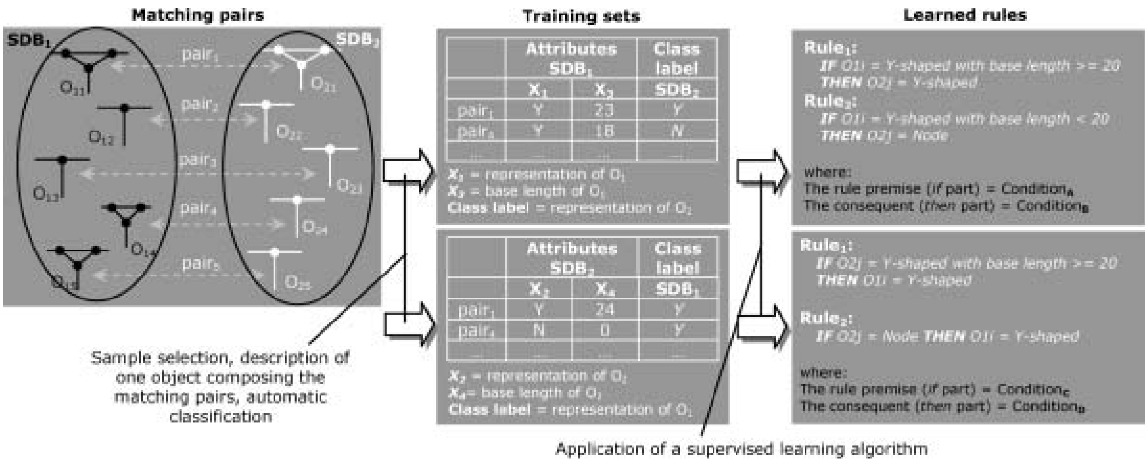
not. The expert takes the specifications into account to fix the class label for the examples. However, they also admit a tolerance on the thresholds encountered in the specifications because they know by experience that this tolerance is introduced during the data capture. This means that the rules obtained with machine learning will not be exactly the same as those existing in the specification documents. In this case, they reflect the expert's knowledge.

The direct classification approach is well adapted to determine the rules of evaluation. It is simple and easy to implement. However, it is well known in the machine-learning community that if the way to describe examples is complex (e.g. with many attributes describing each example), the required training set size may be large for good performances. Acquiring such large training sets interactively may be difficult and costly. For that reason, we argue that this approach can probably not be followed in practice all the time. That is the reason why, even if this approach is theoretically interesting, we did not apply it during the experiments described in section 6. An application can be found in Sheeren (2003).

5.2 *Learning predictive rules from data*

Another approach may be used to acquire knowledge from data, namely the predictive approach. As we saw previously, the predictive approach is based on the definition of the conditions the representation of the objects in one database have to respect, these conditions depending on the representation of the homologous objects in the other database (section 4.2.2). Machine learning can help acquire these conditions automatically. A training example is then composed of attributes that describe the representation of the object in the first database for a given matching pair. The label of the example corresponds to the conditions that the homologous object of the pair must satisfy. If a set of examples is defined in both directions, the learning algorithm can be trained, and two sets of predictive rules can be acquired (figure 10).

Since the class of the examples is defined by one of the two representations of the objects composing the matching pair, the expert is not required to assign it. The training set is directly created at the end of the matching step. So, this approach is particularly fast to implement. In addition, it enables one to use a great number of training examples, as much data as are available, which can greatly improve the quality of the rules and make the approach scalable. The predictive approach thus enables us to overcome the main obstacle mentioned above for the direct classification approach: the intervention of the human expert is not required to



	Attributes SDB$_1$		Class label
	X$_1$	X$_2$	SDB$_2$
pair$_1$	Y	23	Y
pair$_4$	Y	18	N
...	...	...	...

X$_1$ = representation of O$_1$
X$_2$ = base length of O$_1$
Class label = representation of O$_2$

	Attributes SDB$_2$		Class label
	X$_3$	X$_4$	SDB$_1$
pair$_1$	Y	24	Y
pair$_4$	N	0	Y
...	...	...	...

X$_3$ = representation of O$_2$
X$_4$= base length of O$_2$
Class label = representation of O$_1$

Rule$_1$:
IF O1i = Y-shaped with base length >= 20
THEN O2j = Y-shaped
Rule$_2$:
IF O1i = Y-shaped with base length < 20
THEN O2j = Node

where:
The rule premise (*if* part) = Condition$_A$
The consequent (*then* part) = Condition$_B$

Rule$_1$:
IF O2j = Y-shaped with base length >= 20
THEN O1i = Y-shaped

Rule$_2$:
IF O2j = Node *THEN* O1i = Y-shaped

where:
The rule premise (*if* part) = Condition$_C$
The consequent (*then* part) = Condition$_B$

Figure 10. Learning process for acquiring predictive rules.

classify the training examples. This key point makes the predictive approach particularly advantageous.

In this approach, the acquired knowledge reflects the implicit knowledge used during the data capture. The tolerance for the thresholds of the specifications the human operators agree upon can be highlighted, in the same way as in the direct classification approach. But the knowledge extracted in this case is different. The knowledge is closer to the data-capture reality. It reflects the human operator's knowledge and takes several points of view into account, the data being captured by different persons.

This method is particularly interesting for collecting training examples. However, the learning step can be made with more noisy data or, in other words, with examples with a wrong label. Indeed, the label of examples is not checked at the end of the data matching. Consequently, some noisy data (i.e. inconsistencies and matching errors) are selected in the training set. If the data are too noisy, there is a risk of learning rules that do not correspond to the database specifications. From the point of view of the inductive learning, the rules then learned can *overfit* the data.

Since this approach enables a great number of training examples to be used, we make the assumption that, in most cases, the inconsistencies are rare enough not to be taken into account in the inductive process. Only the recurring patterns are learned by the algorithm, and we suppose that they correspond to consistent representations. However, at the end of the process, more attention must be given to the evaluation of the learned rules, in order to ensure that this assumption is indeed verified.

6. Applications

This section illustrates the suggested approach on several concrete application examples using three different vector databases from IGN (the French National Mapping Agency): BDTOPO, BDCARTO and GEOROUTE (figure 11). The databases have been defined according to different specifications, in order to fulfil different application domains and geographical analysis levels. BDTOPO is a highly detailed topographic database. The data are derived from aerial photographs and typically used to produce maps at a 1:25 000 scale. BDCARTO is a geographical decametric database used in particular to produce maps at a scale ranging from

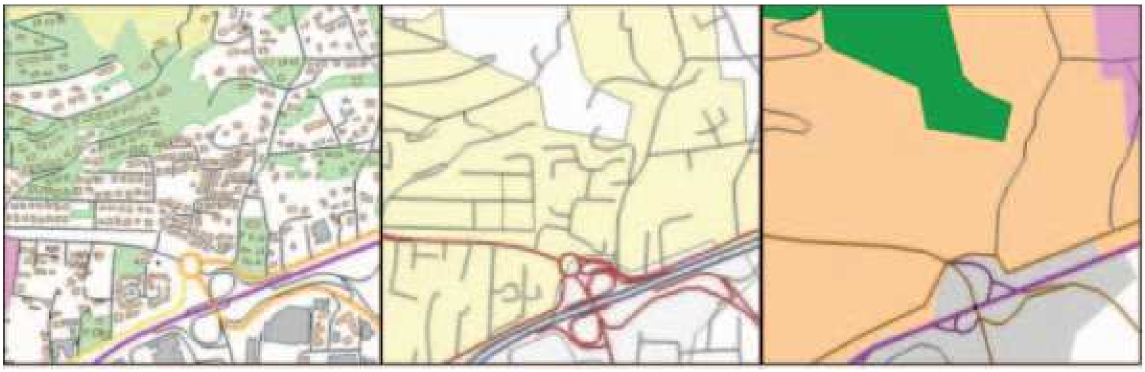


Figure 11. Extract from the three databases we studied: BDTOPO, GEOROUTE and BDCARTO (from left to right).

1:100 000 to 1:250 000. GEOROUTE is a database with a metric resolution specially developed for car-navigation applications. It contains a rich, detailed road network, but the database does not have a cartographic vocation.

The first experiment we made illustrates the detection and the analysis of inconsistencies between geometrical object representations from GEOROUTE and BDCARTO. The application relates to the study of differences between representations of roundabouts, which is one of the possible corresponding object classes between the two databases. The other experiments concern the consistency assessment of attribute values. Several attribute values of objects from BDTOPO and BDCARTO are compared: the voltage of electric lines (numeric attribute) and the nature of orographic points (symbolic attribute). The experiments illustrate how rules for assessing consistency can be derived from data with data mining. They also indicate the interest and limits of the approaches.

6.1 *Inconsistency between geometrical representations*

This first experiment is related to the comparison of roundabout representations from GEOROUTE and BDCARTO. The study area is situated in an urban area near Paris (France) and has an extent of about 4000 km^2. It represents approximately 45 000 road segment objects in GEOROUTE against 14 200 in BDCARTO. In both databases, the road network is described by linear road segments and punctual road nodes.

In the following subsections, we mainly concentrate on the implementation of the MACO method. We briefly describe the enrichment step of MECO, but a more detailed illustration of this method can be found in Sheeren (2005).

6.1.1 Data. In order to present what each database exactly contains and to determine the differences that are likely to appear in the data, we provide some information about the representations of the roundabout object in both databases.

In GEOROUTE, the information relating to the roundabout objects is embedded in the description of two classes: 'Road Node' (point object) and 'Complex Crossroads' (polygon object). A road node corresponds to a road segment extremity in the database. It represents 'crossroads or a change in the circulation conditions in the reality that does not exceed 30 m diameter on the ground'. A 'nature of intersection' attribute specifies the road node type which can take the value 'simple roundabout'. This attribute value is assigned if the crossroads correspond to: 'a place in the road space where roads join each other at the same level. The shape is not exclusively circular. There must have an impassable central reservation and the roads that surround the object must have a gyratory direction' (Georoute 1999). The 'Complex Crossroads' class (polygon object) also refers in GEOROUTE to the roundabout entities in the real world. Complex crossroads can be 'a non-structured traffic area, a large roundabout or arranged crossroads. The minimal extent is 15 m radius. If the extent is lower, the crossroads are modelled as a simple intersection (i.e. a road node)'. This class has also an attribute indicating the nature of the complex crossroads. This attribute can take among others the value 'roundabout'. The definition of a roundabout is then the same as in the 'Road Node' class.

For BDCARTO, the explicit information relating to the roundabout objects is only included in the 'Road Node' class (BDCarto 2001). There exists a *kind of intersection* attribute which can take several values, including 'simple crossroads', 'small roundabout' and 'large roundabout'. The 'simple crossroads' value is assigned if the road node corresponds to a simple intersection, a 'cul-de-sac', arranged crossroads with an extent that does not exceed 100 m, or a roundabout with a diameter lower than 50 metres. Thus, roundabouts are not distinguished from other crossroads when they are small enough. The 'small roundabout' value is allocated for a diameter between 50 and 100 m. Beyond the representation is detailed by means of segments and nodes, and the 'large roundabout' value is given to nodes. The possible representations of roundabouts for both databases with their differences are shown in figure 12.

6.1.2 Extracting implicit spatial objects and properties for assessing consistency. While the roundabout objects can easily be identified visually in the data, no 'Roundabout' class exists in the data. These objects are grouped with objects of another nature into less specific classes ('Road Node' and 'Complex Crossroads'). The concept of a roundabout does not have an explicit existence in BDCARTO and, in particular, for the detailed representation. There is no object with a polygon geometry. A detailed roundabout corresponds only to a set of connected road segments and nodes. In GEOROUTE, the class 'Complex crossroads' includes the detailed roundabouts, but no relationship between the 'Road Node' and 'Road Segment' classes exists. In addition, no attribute relating to the diameter of the objects is stored in the databases. This spatial property is also implicit. So, in order to assess the consistency between the representations of roundabouts, the databases have to be enriched. This was performed during the enrichment step in MECO.

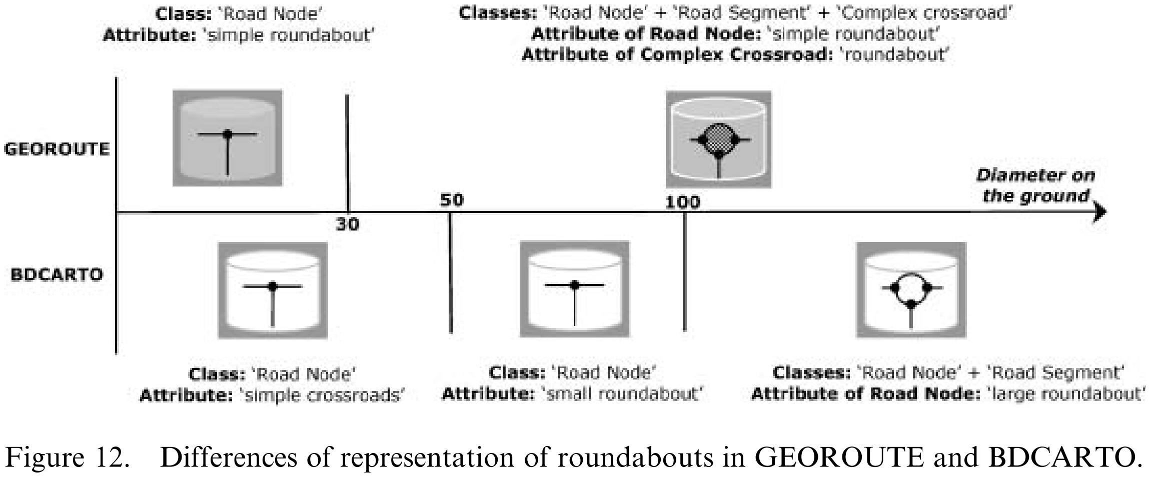

Class: 'Road Node'
Attribute: 'simple roundabout'
Classes: 'Road Node' + 'Road Segment' + 'Complex crossroad'
Attribute of Road Node: 'simple roundabout'
Attribute of Complex Crossroad: 'roundabout'
GEOROUTE
Diameter on the ground
30
50
100
BDCARTO
Class: 'Road Node'
Attribute: 'simple crossroads'
Class: 'Road Node'
Attribute: 'small roundabout'
Classes: 'Road Node' + 'Road Segment'
Attribute of Road Node: 'large roundabout'

Figure 12. Differences of representation of roundabouts in GEOROUTE and BDCARTO.

The enrichment concerns both data and schemata. Explicitly introducing roundabouts in the data supposes the definition of new classes and relations at the schema level; likewise their instantiation (figure 13). The extraction of the roundabouts required several steps. For the simple roundabouts (with a point geometry), the extraction did not present any difficulty, since the simple roundabouts can be selected according to the values of the attribute 'nature of intersection' of the road nodes. For the detailed roundabouts (with a polygon geometry), the extraction of objects in BDCARTO was more complex. First, a topological graph was computed. Then, each face was characterized with Miller's compactness index, the number of nodes associated, and the direction of the cycle. Finally, each face was analysed, and only faces corresponding to a roundabout object were retained (i.e. faces with a compactness higher than 0.95, associated with at least three nodes, and with a gyratory direction). This enrichment was made for both databases. Concerning the BDCARTO, we verified that the objects created were associated with road nodes defined as 'large roundabout'. For GEOROUTE, we verified the correspondence with its complex crossroad. The few encountered internal errors of non-correspondence were detected and dealt with during the consistency assessment itself. An illustration of some roundabouts extracted automatically with this method is given in figure 14.

In a general way, we can see that the enrichment step prepares the data to check its compliance with the specifications, but it also reduces the heterogeneity between the two initial databases. It can help to underline *federative concepts* corresponding to geographical entities defined independently of the representation. The correspondence between these concepts and the objects in each database helps define the unified schema (Gesbert 2004, Uitermark *et al.* 2005).

6.1.3 Knowledge acquisition by data analysis. In this section, we present the knowledge acquired automatically from data to assess consistency. Predictive rules were discovered from a set of training examples. All the pairs computed at the data matching step were selected automatically, and the learning procedure was performed in both directions (to predict conditions that must be respected by both

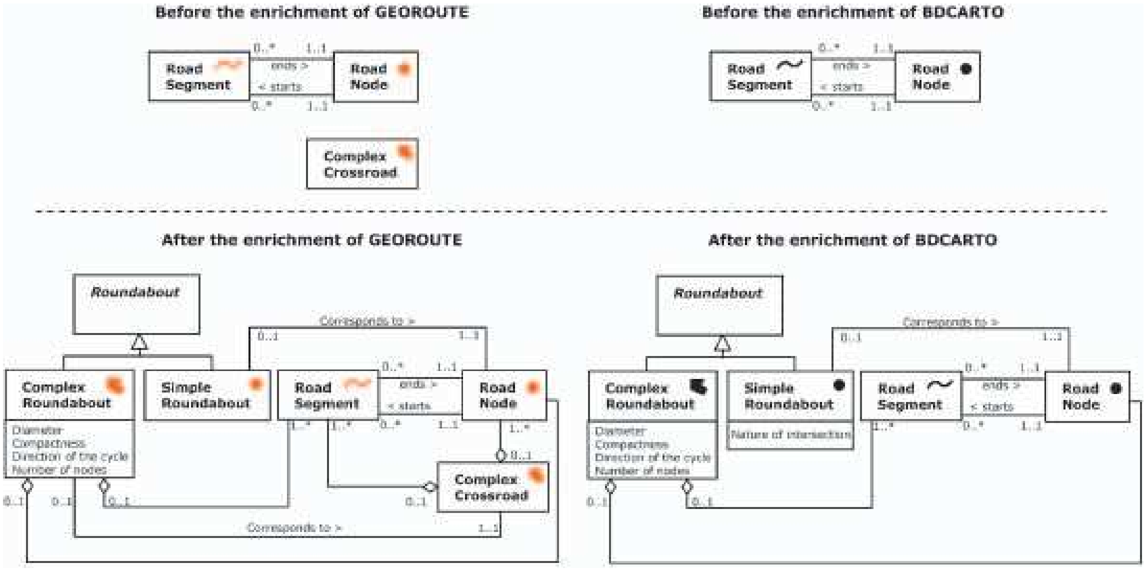


Figure 13. Enrichment of DB schemata of GEOROUTE and BDCARTO. The schemata are expressed in the UML language associated to spatial pictograms used to depict geometry.

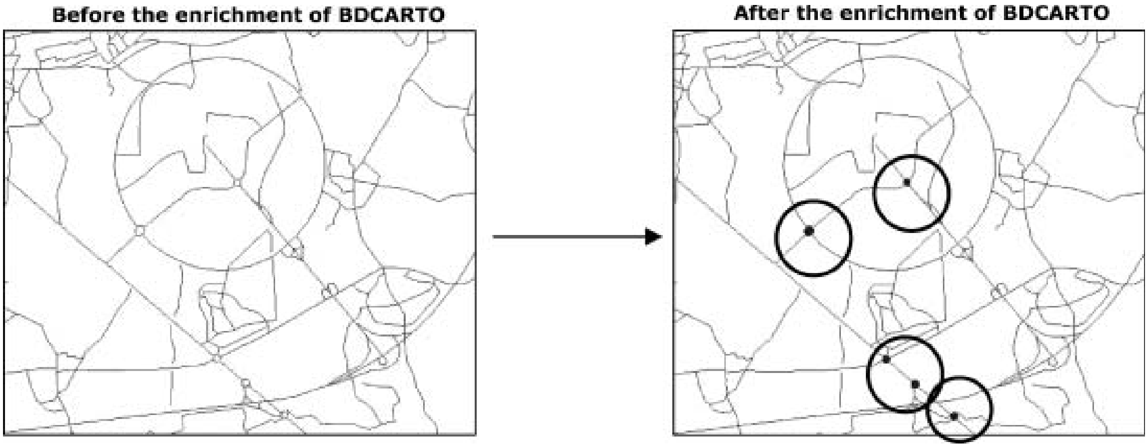

Before the enrichment of BDCARTO
After the enrichment of BDCARTO

Figure 14. Enrichment of data: examples of complex roundabouts extracted automatically in BDCARTO.

BDCARTO and GEOROUTE). Two attributes were retained to predict the conditions that need to be respected by the BDCARTO: the kind of the representation of the roundabouts in GEOROUTE and the diameter length of complex crossroads. The label of the training examples corresponds to the linked representation of BDCARTO, i.e. simple crossroads, a small roundabout, or a large roundabout (table 1).

The rules acquired from the 258 training examples with the C4.5. algorithm are as follows. Prediction of the conditions relating to the BDCARTO representations from the GEOROUTE representations:

R$_1$ If Object$_{\text{GEOROUTE}}$ = 'simple roundabout'
Then Object$_{\text{BDCARTO}}$ must be 'simple crossroads'
R$_2$ If Object$_{\text{GEOROUTE}}$ = 'complex crossroad' and diameter <44
Then Object$_{\text{BDCARTO}}$ must be 'simple crossroads'
R$_3$ If Object$_{\text{GEOROUTE}}$ = 'complex crossroads' and 44<diameter <84
Then Object$_{\text{BDCARTO}}$ must be a 'small roundabout'
R$_4$ If Object$_{\text{GEOROUTE}}$ = 'complex crossroads' and diameter >84
Then Object$_{\text{BDCARTO}}$ must be a 'large roundabout'

All of these rules can be considered relevant. The expression and the number of the learned rules are similar to those that could be deduced from the specifications.

Table 1. Excerpt of the training set enabling to learn the conditions that must be respected by the BDCARTO.

	Attributes		Class
ID	Roundabout representation in GEOROUTE	Diameter length in GEOROUTE	Roundabout representation in BDCARTO
1	Simple roundabout	0.0	Simple crossroads
2	Simple roundabout	0.0	Simple crossroads
3	Complex crossroads	42	Small roundabout
4	Complex crossroads	81.28	Small roundabout
5	Simple roundabout	0.0	Simple crossroads
6	Complex crossroads	91.55	Large roundabout
7	Complex crossroads	70.58	Small roundabout

First, in the presence of a simplified roundabout in GEOROUTE, the only possible consistent representation in BDCARTO is a road node with the attribute value ‘simple crossroads’ (R_1). Then, if there is a detailed representation in GEOROUTE, the consistent representation in BDCARTO can be either simplified or detailed according to the diameter (R_2, R_3, R_4). In the first case, the attribute of the node can take the ‘simple crossroads’ value (diameter <44) or ‘small roundabout’ value ($44<$ diameter <84). In the other case, the diameter length should be higher than 84 m. However, when comparing with the textual specifications, the learned thresholds relating to the diameter length are different. We discovered a threshold of 44 m instead of 50 m in the specifications and a threshold of 84 m instead of 100.

Discovering these differences underlines the interest in the learning approach. Data mining enables us to extract not only the correspondences between object classes from data instances but also the imprecision of the specifications. In general, those in charge of the data capture do not strictly respect the thresholds fixed in the textual specifications. The capture constraints are not strict rules but rather guidelines. The thresholds are defined to give an order of magnitude. Thus, we determined the implicit knowledge used during the data capture obtained from aerial photographs. Learning these differences is particularly interesting because the learned thresholds are closer to the data and, thus, the assessment of consistency is closer to database reality. In addition, it enables us to learn part of the human operator’s know-how. In our case study, we can notice, for instance, that the operators detail the roundabouts in BDCARTO more often than they should.

The results obtained only concern the conditions that should be respected by the BDCARTO. However, the predictive approach requires rules to be defined in both directions. Thus, we also trained the C4.5 algorithm to discover the conditions that have to be respected by GEOROUTE. The same training set was used, but this time, the examples have the form of attributes describing the roundabout representation in BDCARTO, and the label of the examples describes the roundabout representation in GEOROUTE. Three rules were discovered:

R_1 If $\text{Object}_{\text{BDCARTO}}$ = ‘simple crossroads’
Then $\text{Object}_{\text{GEOROUTE}}$ must be a ‘simple roundabout’
R_2 If $\text{Object}_{\text{BDCARTO}}$ = ‘small roundabout’
Then $\text{Object}_{\text{GEOROUTE}}$ must be ‘complex crossroads’
R_3 If $\text{Object}_{\text{BDCARTO}}$ = ‘large roundabout’
Then $\text{Object}_{\text{GEOROUTE}}$ must be ‘complex crossroads’

Among these rules, the first is not complete. According to the specifications, if the BDCARTO object is simple crossroads, the roundabout object in GEOROUTE can be either a simple roundabout or a detailed roundabout. Thus, for this case, the rule has been revised interactively, which shows the interest in using a symbolic learning algorithm. Learned rules are understandable and thus can be interactively revised if necessary. The other rules are correct. However, the class value could be more precise by indicating the diameter to respect. This information has not been learned for the complex crossroads in GEOROUTE. The symbolic algorithm we use does not admit numeric continuous values for the class, and a discretization of these values is not well adapted because the results may be very sensitive to the thresholds fixed for discretizing: an a priori choice of these thresholds is difficult. So, for the last two rules, it is not possible to learn the thresholds for the diameter length.

Even if it is difficult to learn thresholds for continuous values, we can consider that machine learning gives a precious assistance in knowledge acquisition. These experiments show that, in addition to rules for assessing consistency, it is possible to discover the imprecision of the database specifications and determine in a nonarbitrary way an 'ε' value automatically that is related to the thresholds fixed in the documents. However, the interest in using machine learning is broader. The data analysis also enables us to enrich the database specifications by comparing the representations. As we mentioned before, one of the conditions to capture a roundabout in GEOROUTE is to have an impassable central reservation on the ground. In the BDCARTO, no information exists for this criterion. If the results show that only a few inconsistencies appear between the detailed roundabout representations of the two databases, we could make the assumption that the existence of a central reservation on the ground is also a condition to capture the roundabouts in BDCARTO. If this is the case, the textual specifications of BDCARTO could be enriched. Thus, analysing and comparing the representations in this way can also help discover implicit specifications.

6.1.4 Applying rules and detecting inconsistencies. The application described above has been entirely tested and implemented within the MECOLib prototype. The architecture of our prototype is built on three main widely used open-source components: the experimental GeOxygene open-source GIS platform (Badard and Braun 2004), the Java Expert System Shell—JESS (Friedman-Hill 2003), and the WEKA data-mining software (Witten and Frank 2005). GeOxygene provides an object-oriented data model in Java which implements the OGC specifications and ISO standards related to the geographic domain. The GeOxygene users can define their database schemata and applications from this extensible model. All the data instances are manipulated in the object-oriented paradigm, but they are stored in a relational DBMS which can be either Oracle Spatial or PostGIS. A mapping exists between the relational tables and the Java classes. This is supported by the OJB library (Apache Object Relational Bridge). GeOxygene is linked to the other components of MECOLib (i.e. JESS and WEKA) via Java APIs.

Graphical user interfaces (GUI) have been specifically developed in the prototype to analyse the matching pairs and give an overview of the evaluation results. These are illustrated in figure 15.

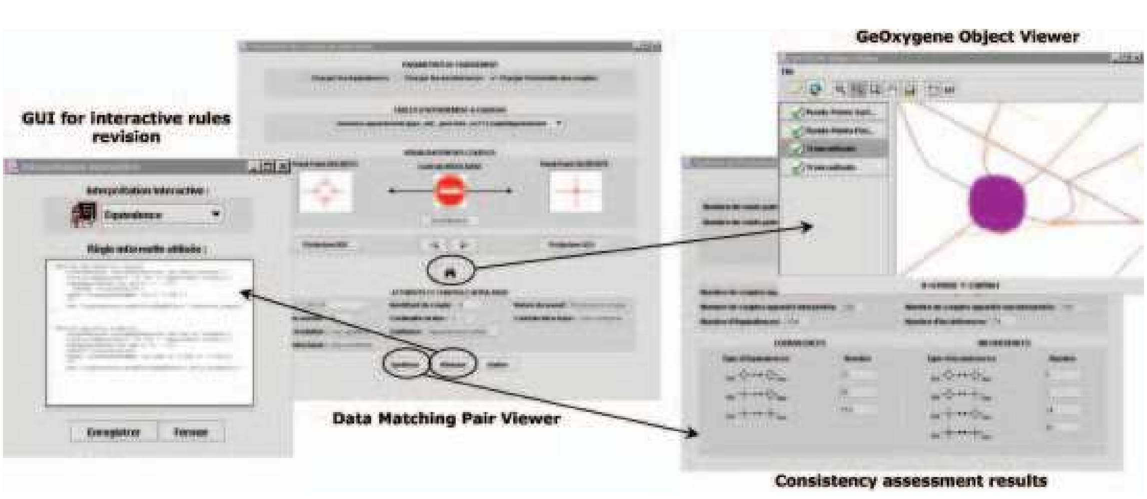

GeOxygene Object Viewer
GUI for interactive rules revision
Data Matching Pair Viewer
Consistency assessment results

Figure 15. Graphical user interfaces of the MECOLib prototype.

Using these tools, a human operator can manually select the inconsistent correspondences and correct the incorrect object representations before the data integration itself. If necessary, they can also revise the learned evaluation rules introduced manually in the expert system. This revision can be assisted by the expert system itself that can explain how it led to a decision (i.e. showing the rules that were used) when an incorrect decision was made and an error detected. This is one of the advantages of using an expert system.

Table 2 shows all the consistent and inconsistent representations of the different correspondence categories which appeared after the implementation and the activation of the learned rules in the expert system. Twenty-nine per cent of inconsistencies were found in the data, against 71% of consistent pairs. These values were computed at the end of the inter-DB control but also take into account the results of the intra-DB control to explain the origin of the inconsistencies (table 3). The inconsistencies detected have been interactively checked one by one, and all turned out to be actual inconsistencies, which encourages the approach.

If an inconsistency exists between two simplified representations (i.e. two points), the error can come from either the BDCARTO (the road node is classified as 'small roundabout' instead of 'simple crossroads') or GEOROUTE (the representation is simplified, although it should be detailed). In this case, all we can do is notice the inconsistency, but it is not possible to specify in which database the error exists (table 3, see the fourth case). On the contrary, if the representation is simplified in the BDCARTO (i.e. the road node is classified as 'simple crossroads'), and if this one is detailed in GEOROUTE (with a diameter larger than 50 m), one can make the assumption that the error lies in BDCARTO. The diameter was probably not overestimated compared with the reality, since the data capture derived from aerial photographs in GEOROUTE (table 3, second case).

The experiment presented here involved approximately 260 roundabouts. However, since both the data matching and the predictive approach are fully automated, the method can be also used for large databases with many object classes and many instances. Experiments have been performed in that sense and in particular to check the consistency between attribute values of road segments (Sheeren 2005). From about 45 000 road segment objects in GEOROUTE against 14 200 in BDCARTO, we computed approximately 7000 matching pairs, and all of these pairs were used to learn predictive rules. So, conclusions about the matching of

Table 2. Results of the inter-database control applied on the set of the 258 extracted roundabouts.

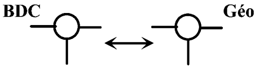
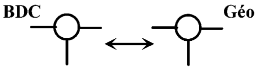
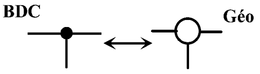
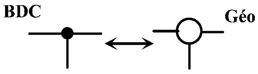
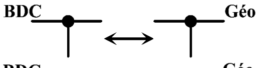
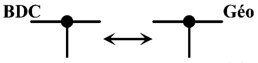
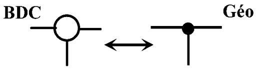
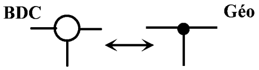
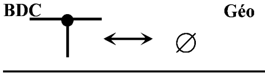
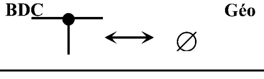
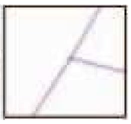
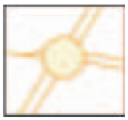
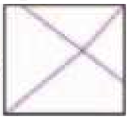
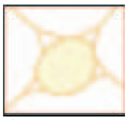
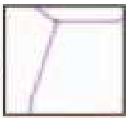
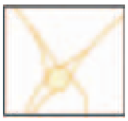
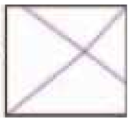
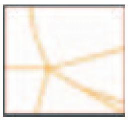
Consistent pairs: 184 matching pairs (71%)		Inconsistent pairs: 74 matching pairs (29%)	
Type	No. of matching pairs	Type	No. of matching pairs
	10 (5.5%)		6 (8%)
	56 (30.5%)		24 (32.5%)
	118 (64%)		40 (54%)
	Impossible by definition		1 (1.5%)
	Impossible by definition		3 (4%)

Table 3. Examples of consistent and inconsistent representations.

BDCARTO	GEOROUTE	Results of the Intra-DB control of BDCARTO	Results of the Intra-DB control of GEOROURTE	Results of theInter-DB control
1)  Simple crossroad	 Ø=40 m	Simple representation → intra-DB not applied	Consistent diameter value → potentially in compliance with the specifications	Consistent representations
2)  Small roundabout	 Ø=39 m	Simple representation → intra-DB not applied	Consistent diameter value → potentially in compliance with the specifications	Inconsistent representations → BDCARTO is probably not in compliance with the specifications
3)  Simple crossroad	 Ø=15 m	Simple representation → intra-DB not applied	Inconsistent diameter value→ internal error	Inconsistent representations → GEROUTE is not in compliance with the specifications
4)  Small roundabout	 Simple roundabout	Simple representation → intra-DB not applied	Simple representation → not applied	Inconsistent representations

the corresponding classes and the rules derived from it can be drawn using a great number of training examples, i.e. using as many matching pairs as available. This is what makes the predictive learning approach particularly interesting and scalable.

6.2 *Inconsistency between attribute values*

The first experiment presented an application example related to the detection of inconsistencies between geometrical representations. In this section, we show some experiments on the comparison of attribute values. This is another kind of multi-representation: real-world entities may be represented in two databases by different objects with different attribute values, either consistent or not. For the sake of clarity, we will consider in these experiments that only one-to-one matching links have been detected, i.e. one object from the first database (here the BDTOPO) is linked to only one object from the second database (the BDCARTO). However, in practice, this is not the case. A difference in granulometry exists between the linear object representations considered which results in a *fragmentation* conflict (Parent *et al.* 1996). But in order to reduce the spatial heterogeneity between data and to facilitate the comparison, the geometrical objects of BDCARTO have been segmented according to the representation of the BDTOPO objects. In this way, we can directly compare the attribute values of the objects. This transformation, which has no influence on the conclusions of the experiments, is another illustration of the enrichment step proposed in MECO.

The first experiment concerns the voltage of electric lines in the two databases (Mustière 2006). For this attribute, the possible values are the same in both spatial databases. But in spite of these apparent similarities confirmed by the textual

Table 4. Confusion matrix for the voltage value of matched electric lines.

		BDCARTO				
		63 kV	90 kV	150 kV	225 kV	400 kV
BD	63 kV	169	28	0	2	0
TOPO	90 kV	8	12	0	0	0
	150 kV	0	0	17	0	0
	225 kV	0	0	0	32	0
	400 kV	0	0	0	0	9
	Unknown	1	0	0	2	0

specifications, the data analysis may highlight differences. This experiment shows once again the implicit fuzziness of the specifications and the data, and the interest of data mining to discover rules for assessing consistency.

The confusion matrix between attribute values of matched electric lines is shown in table 4. In each cell, the number of matched pairs of objects that have some given values in the two databases is computed. For instance, there are eight pairs of objects with a BDTOPO voltage of 90 kV that correspond with a BDCARTO voltage of 63 kV. Notice that the automated matching results in this experiment as well as in the following one are fully efficient and that all the matching pairs have been interactively checked.

Three important points are shown in this matrix. First, two matching pairs are clearly inconsistent: those with the BDTOPO voltage of 63 kV associated with those with the BDCARTO voltage of 225 kV, but the machine-learning algorithm C4.5 easily determined from this matrix the rule ‘if the BDCARTO voltage is 225 then the BDTOPO voltage is 225’. It thus rightly considered that the two pairs did not fulfil these rules appearing as noise and did not take them into account when inducing learned rules. These two inconsistencies will then be easily detected when applying learned rules.

The second important point is the relatively frequent confusion between 63 kV and 90 kV. If we had only followed textual specifications, we would have determined rules ‘if the BDCARTO voltage is 63 (respectively 90) then the BDTOPO voltage should be 63 (resp. 90)’. Actual data clearly show that the frontier between these two values is not so sharp, maybe because these values are actually approximated and estimated. Thus, confusions between 63 and 90 kV can hardly be thought of as inconsistencies. They rather reflect the implicit fuzziness of the specifications and the data. Unfortunately, most basic machine-learning tools do not deal very well with these cases; they do not provide the expected rule ‘if the BDCARTO voltage is 63 or 90 then the BDTOPO voltage should be 63 or 90’, because they do not foresee rules conclusions with a disjunction such as ‘63 or 90’. Two approaches however may overcome this problem: one can either build an ad hoc method for simple confusion matrices or use more complex learning tools that allow the automatic grouping of class values like some learning methods do (Ganascia *et al.* 1993).

Another important point concerns the high voltage 400 kV lines. Unfortunately, the machine-learning algorithm did not produce the expected rule ‘if the BDCARTO voltage is 400 then the BDTOPO voltage should be 400’ because the number of examples to support this rule were not suffient. Similarly, the unknown values have not been taken into account. Simply changing some parameters of the learning algorithm did solve the problem in this experiment. But this highlights the fact that the frontier between inconsistencies on the one hand and rare but consistent combinations on the other hand is hard to determine automatically. This is certainly the main limit of our approach that relies on data mining. This also suggests that learned rules should be checked systematically and confirms our decision to use symbolic machine-learning tools that produce readable rules rather than numeric approaches such as ANN or SVM. Indeed, these approaches do not provide an interpretable model, and so no expertise can be used to analyse it.

The second experiment is similar to the previous one but it has a symbolic attribute to describe the nature of the object rather than a numeric attribute describing a property. We compared two classes supposed to represent main orographic points (summits, passes, gorges ...). Results are shown in table 5.

Similar conclusions to the previous conclusions can be drawn from this experiment. First, some actual inconsistencies may be detected, like that between a *Summit* and a *Versant*. Second, the matrix shows some confusion that would not have been directly seen in the textual map specifications, even if it seems natural and reflects the fuzziness of the information: objects qualified as *Peak* in BDTOPO may correspond to objects qualified as *Peak* in BDCARTO but also to objects qualified as *Crest, summit or mountain.* Similarly, objects in BDTOPO qualified as *Summit, crest or hill* may correspond to objects qualified as *Crest, summit or mountain* in BDCARTO, but also to objects named *Peak*. Third, some attribute values are very rarely encountered in the sample data (or even never in the case of volcanoes), and so data-mining tools can hardly discover relations for these types of attributes.

7. Conclusions and outlook

In this paper, we theoretically defined inconsistencies as differences between databases that cannot be explained by their respective specifications. We also alleged

Table 5. Confusion matrix for the nature value of matched orographic points.

		BDCARTO										
		Cape	Cirque	Pass	Versant	Dune, Isthmus, Beach	Peak	Plain, plateau	Rocks, Escarpment	Crest, summit, mountain	Valley, Gorges	Volcano
BD	Cape, headland	3	0	0	0	0	0	0	0	0	0	0
TOPO	Cirque	0	2	0	0	0	0	0	0	0	0	0
	Pass, passage	0	0	91	0	0	0	0	0	1	0	0
	Hillside, cliff	0	0	0	3	0	0	0	1	0	0	0
	Dune, beach	0	0	0	0	2	0	0	0	0	0	0
	Peak	0	0	0	0	0	60	0	0	41	0	0
	Plain, plateau	0	0	0	0	0	0	2	0	0	0	0
	Rocks	0	0	0	0	0	0	0	3	0	0	0
	Summit, crest, hill	0	0	0	1	0	7	0	0	104	0	0
	Valley	0	0	0	0	0	0	0	0	0	8	0
	Crater	0	0	0	0	0	0	0	0	0	0	0

that in practice, detecting these inconsistencies requires relying on a complex body of knowledge. Knowledge representation and acquisition are thus key issues for consistency assessment.

We proposed a bottom-up approach to acquire the necessary knowledge to partially automate the identification of inconsistencies. As they are full of important information, textual specifications provided by data producers could be exploited. Nevertheless, this requires translating textual specifications into formal models and computational languages, which is a natural language-processing task that is known to be very difficult. As an alternative approach to acquire this knowledge, we propose automatically inferring the specifications from the data instances themselves. This step resorts in practice on using data-mining tools.

We also proposed two ways to represent the necessary knowledge by means of production rules. The first one, called the 'direct classification approach', consists in organizing the rules in the form of 'if condition(object in DB1, corresponding object in DB2) then the objects are consistent or not'. The second one, called the 'predictive approach', consists in organizing the rules in the form of 'if condition$_1$(object in DB1) then the corresponding object in DB2 has to respect condition$_2$ to be consistent'. A theoretical analysis of these approaches, as well as some experiments performed on actual data, enables us to underline the differences between them, as summarized in table 6 (see below). Depending on the chosen approach, the knowledge-acquisition process may be a more or less difficult task. However, the most important difference between these two approaches may well be that they elicit a different type of knowledge, reflecting either the human expert's knowledge (i.e. the supervisor) or the knowledge embedded in the data (i.e. the persons that capture the spatial data). Since the data are captured by several human operators, the rules learned following the predictive approach are probably more objective and closer to the data-capture reality. In this respect, the data-mining approach thus appears to be a promising approach, either for learning specifications from data when they are too complex, difficult to acquire, or do not exist, or for studying the differences between textual specifications and those actually used to produce the data.

Due to the existing implicit fuzziness of both the specifications and the data, a more flexible mode of classification ought to be adopted in the future. Labelling matching pairs only as either consistent or inconsistent is probably too restrictive. An extension of the approach that captures uncertainty in the classification should

Table 6. Two knowledge representation and acquisition approaches proposed.

	Direct classification	Predictive approach
Knowledge-acquisition process	Semi-automatic: rules are automatically learned but examples have to be defined, collected and classified by an expert.	Automatic: rules are automatically learned and examples are automatically built with data-matching tools, but an expert still needs to define the form of the examples
Acquired knowledge	It reflects the human domain expert's knowledge (i.e. the supervisor)	It reflects knowledge "embedded" in the data (i.e. the human operator's knowledge who captured the data)
In practice	Time-consuming because the expert has to collect a lot of examples by hand	Fast to develop, but special attention has to be paid to the rules automatically acquired, because they may have been generated from incorrect or noisy examples

be envisaged (Comber *et al.* 2004). In addition, the extraction of rules from data should also be studied using other learning algorithms capable of learning more complex or efficient rules from noisy data. Another perspective is the application of the method to raster data (Fritz and See 2005).

The approach we proposed is a starting-point for further investigations regarding the data-driven schemata integration (Duckham and Worboys 2005, Volz 2005). Our approach also opens up new prospects for the enrichment of the specifications. In a more general way, this is a first step towards the automatic extraction of metadata deriving from spatial data. Furthermore, we believe that combining top-down and bottom-up approaches is the key to successfully integrating spatial databases in a consistent way. Textual specifications should be used to automatically assess the relevance of the learned rules and complete them when required. This would enable us to avoid the interactive validation and revision steps included in our method. In that sense, a rich formal model was recently defined to describe spatial database specifications (Gesbert 2004, 2005). This model relies on an ontology of the geographic world and describes specifications as links between the ontology and the classes of the database schema. These links themselves rely on a formal description of typical constraints encountered in the databases like geometric constraints (ex: 'an house is captured only if it is bigger than 100 m^2'), topological constraints ('roads should be connected'), nature constraints ('only paved roads are captured'), and relational constraint ('a small path is captured only if it leads to an important building') (Mustière *et al.* 2003). The automatic translation of the textual specifications into this formal model defined is currently being explored.

References

AHLQVIST, O., KEUKELAAR, J. and OUKBIR, K., 2003, Rough and fuzzy geographical data integration. *International Journal of Geographical Information Science*, **17**, pp. 223–234.

BADARD, T. and BRAUN, A., 2004, OXYGENE: A Platform for the development of interoperable geographic applications and web services. In *Proceedings of the 15th International Workshop on Database and Expert Systems Applications*, (DEXA'04), pp. 888–892 (New York: IEEE Press). Available online at: http://oxygene-project.sourceforge.net/ (accessed 27 February 2008).

BALLEY, S., PARENT, C. and SPACCAPIETRA, S., 2004, Modeling geographic data with multiple representations. *International Journal of Geographical Information Science*, **18**, pp. 329–354.

BATINI, C., LENZERINI, M. and NAVATHE, S.B., 1986, A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, **18**, pp. 323–364.

BDCARTO, 2001, Specifications de contenu de la BDCarto® version 2.0, IGN, St Mandé.

BEERI, C., KANZA, Y., SAFRA, E. and SAGIV, Y., 2004, Object fusion in Geographic Information Systems. In *Proceedings of the 30th VLDB Conference*, pp. 816–827.

BONNETT, R., RODRIGUEZ-BACHILLER, A. and GLASSON, J., 2004, Expert systems and geographic information systems for impact assessment, pp. 400 (CRS press).

BRANKI, T. and DEFUDE, B., 1998, Data and metadata: two-dimensional integration of heterogeneous spatial databases. In *Proceedings of the 8th International Symposium on Spatial Data Handling (SDH'98)*, pp. 172–179.

CALVANESE, D., DE GIACOMO, G., LENZERINI, M., NARDI, D. and ROSATI, R., 2001, Data integration in data warehousing. *International Journal of Cooperative Information Systems*, **10**, pp. 237–271.

COCKROFT, S., 1997, A taxonomy of spatial data intergrity constraints. *GeoInformatica*, **1**, pp. 327–343.

COMBER, A.J., FISHER, P. and WADSWORTH, R., 2004, Integrating land cover data with different ontologies: identifying change from inconsistency. *International Journal of Geographical Information Science*, **18**, pp. 691–708.

DAVID, J.-M., KRIVINE, J.-P., and SIMMONS, R. (Eds), 1993, *Second generation Expert Systems* (Berlin: Springer).

DEVOGELE, T., 1997, Processus d'intégration et d'appariement de bases de données Géographiques. Application à une base de données routières multi-échelles. PhD thesis in Computer Science, University of Versailles (in French).

DEVOGELE, T., PARENT, C. and SPACCAPIETRA, S., 1998, On spatial database integration. *International Journal of Geographical Information Science*, **12**, pp. 335–352.

DUCKHAM, M. and WORBOYS, M., 2005, An algebraic approach to automated geospatial information fusion. *International Journal of Geographical Information Science*, **19**, pp. 537–557.

EGENHOFER, M.J., CLEMENTINI, E. and DI FELICE, P., 1994, Evaluating inconsistencies among multiple representations. In *Proceedings of the 6th International Symposium on Spatial Data Handling (SDH'94)*, pp. 901–920.

EL-GERESY, B.A. and ABDELMOTY, A.I., 1998, A qualitative approach to integration in spatial databases. In *Proceedings of the 9th International Conference on Database and Expert Systems Applications (DEXA'98)*, Lecture Notes in Computer Science Vol. 1460, Springer, pp. 280–289.

FEIGENBAUM, E.A., 1981, Expert systems in the 1980s. In *State of the Art Report on Machine Intelligence*, A. Bond, (Ed) (Maidenhead, UK: Pergamon-Infotech).

FONSECA, F.T., DAVIS, C.A. and CÂMARA, G., 2003, Bridging ontologies and conceptual schemas in geographic information integration. *GeoInformatica*, **7**, pp. 355–378.

FRIEDMAN-HILL, E., 2003, *Jess in Action, Java Rule-based Systems* (Greenwich, CT: Manning Publications).

FRIIS-CHRISTENSEN, A., 2003, Issues in the conceptual modelling of geographic data. PhD thesis in Computer Science, University of Aalborg.

FRITZ, S. and SEE, L., 2005, Comparison of land cover maps using fuzzy agreement. *International Journal of Geographical Information Science*, **19**, pp. 787–807.

GANASCIA, J.-G., THOMAS, J. and LAUBLET, P., 1993, Integrating models of knowledge and machine learning. In *Proceedings of the European Conference on Machine Learning (ECML'93)*, pp. 396–401.

GEOROUTE, 1999, *Specifications de contenu de Géoroute®*, version 2.5 (St Mandé: IGN).

GESBERT, N., 2004, Formalisation of geographical database specifications. In *Proceedings of the 8th East European Conference on Advances in Databases and Information Systems (ADBIS'04)*, pp. 202–211.

GESBERT, N., 2005, Formalisation des spécifications de bases de données géographiques en vue de leur intégration. PhD thesis in Computer Science, University of Marne-La-Vallée (in French).

GOMBOŠI, M., ŽALIK, B. and KRIVOGRAD, S., 2003, Comparing two sets of polygons. *International Journal of Geographical Information Science*, **17**, pp. 431–443.

GOODCHILD, M., and JEANSOULIN, R. (Eds), 1998, *Data Quality in Geographic Information: from Error to Uncertainty* (Paris: Hermes).

HAGEN-ZANKER, A., STRAATMAN, B. and ULJEE, I., 2005, Further developments of a fuzzy set map comparison approach. *International Journal of Geographical Information Science*, **19**, pp. 769–785.

HAMPE, M. and SESTER, M., 2002, Real-time integration and generalization of spatial data for mobile applications. In *Geowissenschaftliche Mitteilungen, Maps and the Internet*, pp. 167–175 (Vienna: Heft).

HAUNERT, J.-H., 2005, Link based conflation of geographic datasets. In *Proceedings of the 8th ICA Workshop on Generalisation and Multiple Representations*. Available online at: http://ica.ign.fr (accessed 27 February 2008).

Kilpeläinen, T., 2000, Knowledge acquisition for generalization rules. *Cartography and Geographic Information Science*, **27**, pp. 41–50.

Leclercq, E., Benslimane, D. and Yétongnon, K., 1999, ISIS: A semantic mediation model and an agent based architecture for GIS Interoperability. In *Proceedings of the International Database Engineering and Applications Symposium (IDEAS'99)*, pp. 87–91.

Lemarié, C. and Badard, T., 2001, Cartographic database updating. In *Proceedings of the 20th International Cartographic Conference (ICC 2001)*, vol. 2, pp. 1376–1385.

Leung, Y. and Leung, K.-S., 1993, An intelligent expert system shell for knowledge-based geographical information systems: 2. Some applications. *International Journal of Geographical Information Systems*, **7**, pp. 201–213.

Litwin, W., Mark, L. and Roussopoulos, N., 1990, Interoperability of multiple autonomous databases. *ACM Computing Surveys*, **22**, pp. 267–293.

Musen, M., 1993, An overview of knowledge acquisition. In J.M. David, J.P. Krivine and R. Simmons (Eds). *Second Generation Expert Systems*, pp. 405–427 (Berlin: Springer).

Mustière, S., 2005, Cartographic generalization of roads in a local and adaptive approach: A knowledge acquisition problem. *International Journal of Geographical Information Science*, **19**, pp. 937–955.

Mustière, S., 2006, Results on experiments on automated matching of networks. In *Proceedings of the ISPRS Workshop on Multiple Representation and Interoperability of Spatial Data*, Hanover, pp. 92–100.

Mustière, S., Gesbert, N. and Sheeren, D., 2003, A formal model for the specifications of geographic databases. In S. Levachkine, J. Serra and M. Egenhofer (Eds). *Proceedings of the International Workshop on Semantic Processing of Spatial Data (Geopro'2003)*, Mexico, pp. 152–159.

Mustière, S. and van Smaalen, J., 2007, Databases requirements for generalisation and multiple representations. In W. Mackaness, A. Ruas and T. Sarjakoski (Eds). *The Generalisation of Geographic Information: Models and Applications* (Amsterdam: Elsevier).

Mustière, S., Zucker, J.-D. and Saitta, L., 2000, An abstraction-based machine learning approach to cartographic generalisation. In *Proceedings of the 9th International Symposium on Spatial Data Handling (SDH'2000)*, Beijing, Section 1a, pp. 50–63.

Mitchell, T.M., 1997, *Machine Learning* (Singapore: McGraw-Hill).

Paiva, J.A., 1998, Topological equivalence and similarity in multi-representation geographic databases. PhD thesis, University of Maine.

Parent, C. and Spaccapietra, S., 2000, Database integration: the key to data interoperability. In M. Papazoglou, S. Spaccapietra and Z. Tari (Eds). *Advances in Object-Oriented Data Modelling*, pp. 221–253 (Cambridge, MA: MIT Press).

Parent, C., Spaccapietra, S. and Devogele, T., 1996, Conflicts in spatial database integration. In *Proceedings of the 9th Conference on Parallel and Distributed Computing Systems (PDCS'96)*, pp. 772–778.

Park, J., 2001, Schema integration methodology and toolkit for heterogeneous and distributed geographic databases. *Journal of the Korea Industrial Information Systems Society*, **6**, pp. 51–64.

Quinlan, J.R., 1993, *C4.5: Programs for Machine Learning* (San Francisco: Morgan Kaufmann).

Rodriguez, A., 2005, Inconsistency issues in spatial databases. In L. Bertosi, T. Hunter and T. Schaub (Eds). *Inconsistency Tolerance*, Lecture Notes in Computer Science Vol. 3300, pp. 237–269.

Rodriguez, A. and Egenhofer, M., 2004, Comparing geospatial entity classes: an asymmetric and context-dependent similarity measure. *International Journal of Geographical Information Science*, **18**, pp. 229–256.

Servigne, S., Ubeda, T., Puricelli, A. and Laurini, R., 2000, A Methodology for spatial consistency improvement of geographic databases. *GeoInformatica*, **4**, pp. 7–34.

SESTER, M., 2000, Knowledge acquisition for the automatic interpretation of spatial data. *International Journal of Geographical Information Science*, **14**, pp. 1–24.

SESTER, M., ANDERS, K.-A. and WALTER, V., 1998, Linking objects of different spatial data sets by integration and aggregation. *GeoInformatica*, **2**, pp. 335–358.

SHEEREN, D., 2003, Spatial databases integration: interpretation of multiple representations by using machine learning techniques. In *Proceedings of the 21st International Cartographic Conference (ICC'03)*, pp. 235–245.

SHEEREN, D., 2005, Méthodologie d'évaluation de la cohérence inter-représentation pour l'intégration de bases de données spatiales. PhD thesis, University of Paris 6 (in French).

SHEEREN, D., MUSTIÈRE, S. and ZUCKER, J.-D., 2004a, How to integrate heterogeneous spatial databases in a consistent way? In A. Benczur, J. Demetrovics and G. Gottlob (Eds). *Advances in Databases and Information Systems* (ADBIS'04), Lecture Notes in Computer Science, Vol. 3255, pp. 364–378.

SHEEREN, D., MUSTIÈRE, S. and ZUCKER, J.-D., 2004b, Consistency assessment between multiple representations of geographical databases: a specification-based approach. In P. Fisher (Ed). *Developments in Spatial Data Handling (SDH'04)*, pp. 617–628 (Berlin: Springer).

SHETH, A. and LARSON, J., 1990, Federated database systems for managing distributed, heterogeneous and autonomous databases. *ACM Computing Surveys*, **22**, pp. 183–236.

STRAUCH, J., SOUZA, J. and MATTOSO, M., 1998, A methodology for GIS database integration. In *Proceedings of the IEEE Workshop on Knowledge and Data Engineering Exchange (KDEX'98)*, pp. 151–159.

TOMAI, E., 2006, Towards intensional/extensional integration between ontologies. In *Proceedings of the ISPRS Workshop on Multiple Representation and Interoperability of Spatial Data*, Hanover.

UITERMARK, H., VAN OOSTEROM, P., MARS, N. and MOLENAAR, M., 2005, Ontology-based integration of topographic data sets. *International Journal of Applied Earth Observation and Geoinformation*, **7**, pp. 97–106.

VANGENOT, C., PARENT, C. and SPACCAPIETRA, S., 2002, Modelling and manipulating multiple representations of spatial data. In *Proceedings of the 10th International Symposium on Spatial Data Handling (SDH'02)*, pp. 81–93.

VISSER, H., STUCKENSCHMIDT, H., SCHUSTER, G. and VÖGELE, T., 2002, Ontologies for geographic information processing. *Computers & Geosciences*, **28**, pp. 103–117.

VOLZ, S., 2005, Data driven matching of geospatial schemas. In A.G. Cohen and D.M. Mark (Eds). *International Conference on Spatial Information Theory (COSIT'05)*, Lecture Notes in Computer Science, Vol. 3693, pp. 115–132.

VOLZ, S., 2006, An iterative approach for matching multiple representations of street data. In *Proceedings of the ISPRS Workshop on Multiple Representation and Interoperability of Spatial Data*, pp. 101–110.

WALTER, V. and FRITSCH, D., 1999, Matching spatial data sets: a statistical approach. *International Journal of Geographical Information Science*, **13**, pp. 445–473.

WEIBEL, R., KELLER, S. and REICHENBACHER, T., 1995, Overcoming the knowledge acquisition bottleneck in map generalization: the role of interactive systems and computational intelligence. In A.U. Frank and W. Kuhn (Eds). *Spatial Information Theory—A Theoretical Basis for GIS (COSIT'95)*, Lecture Notes in Computer Science, Vol. 988, pp. 139–156.

WIEDERHOLD, G., 1992, Mediators in the architecture of future information systems. *IEEE Computer*, **25**, pp. 38–49.

WORBOYS, M. and CLEMENTINI, E., 2001, Integration of imperfect spatial information. *Journal of Visual Languages and Computing*, **12**, pp. 61–80.

WITTEN, I.H. and FRANK, E., 2005, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, 2nd edition (San Francisco: Morgan Kaufmann).