GEOLIS : A Logical Information System for Geographical Data

O. Bedel* [1] — **S. Ferré*** — **O. Ridoux*** — **E. Quesseveur****

* *Univ. Rennes 1/IRISA, Equipe LIS*
Campus de Beaulieu
35042 Rennes Cedex FRANCE
{prénom.nom}@irisa.fr

** *RESO, UMR CNRS ESO 6590*
Univ. Rennes 2, Campus Villejean
35 043 Rennes Cedex FRANCE
{prénom.nom}@uhb.fr

ABSTRACT. *Today, the thematic layer is still the prevailling structure in geomatics for handling geographical information. However, the layer model is rigid: it implies partitionning geographical data in predefined categories and using the same description schema for all elements of a layer. Recently, Logical Information Systems (LIS) introduced a new paradigm for information management and retrieval. Using LIS, we propose a more flexible organisation of vectorial geographical data at a thiner level since it is centered on the geographical feature. LIS do not rely on a hierarchical organisation of information, and enable to tightly combine querying and navigation. In this article, we present the use of LIS to handle geographical data. In particular, we detail a data model for geographical features and the corresponding querying and navigation model. These models have been implemented in the GEOLIS prototype, which has been used to lead experiments on real data.*

RÉSUMÉ. *La structuration de l'information géographique en couche thématique est actuellement le modèle d'organisation le plus usité en géomatique. Cependant, ce modèle peut paraître rigide : il impose une partition des données géographiques et un schéma de description fixe par couche. Depuis peu, les Systèmes d'Information Logiques (SIL) offent un nouveau paradigme pour l'organisation et la recherche d'information. Avec les SIL, nous proposons un modèle d'organisation des données géographiques vectorielles plus flexible, centré sur l'entité géographique. Les SIL n'imposent aucune structuration hiérarchique de l'information et permettent de combiner étroitement interrogation et navigation. Dans cet article, nous présentons nos travaux sur l'utilisation des SIL en géographie. Nous détaillons un modèle de données, d'interrogation et de navigation, et nous illustrons son application sur un jeu de données réelles.*

KEYWORDS: *Logical Information Systems, geographical data, navigation, querying.*

MOTS-CLÉS : *Systèmes d'information logiques, données géographiques, requêtes, navigation.*

1. Olivier Bedel benefits from a doctoral grant from Region Bretagne.

1. Introduction

Currently, layer organisation is the prevailling model for handling information in Geographical Information Systems (GIS). This structure gathers geographical data under a common theme e.g., soils, roads, water (Laurini *et al.*, 1992), and has become a standard for handling data. It enables to link a cartographic representation to a particular theme and to produce new information from layers processing, e.g. map algebra (Bruns *et al.*, 1997). However, this structure is rigid as it implies partitioning geographical information in predefined categories, and usually having the same description schema for all the elements of a layer. As a consequence, data belonging to several themes are often duplicated in corresponding layers. Furthermore, the layer model is not really designed to manage relations between objects. If current GIS tools enable to query and to work on data distributed in several layers, pre-processing or repeated operations are often necessary.

On the opposite, the Logical Information System (LIS) model was proposed to avoid the rigidity of hierarchical data systems, and to merge querying facilities (as in databases) and navigation facilities (as in hierarchical filesystems). Logical Information Systems (LIS) offer a new paradigm for information management and retrieval. This paradigm is characterized by the following principles (Ferré *et al.*, 2004):

- information is centered on the objects of interest, i.e., on the entities one wishes to classify and retrieve (e.g., files, bibliographical references, geographical features),
- querying and navigation are tightly combined, so that users can freely mix them in a same search,
- the navigation structure is automatically derived and continuously updated w.r.t. data,
- logic is used in all aspects of object description, querying, and navigation, which provides a uniform and expressive language, as well as automated reasoning capabilities.

In this article, we explore how GIS applications can gain in flexibility by using LIS for handling geographical data. We propose an organisation centered on the geographical feature that avoids the rigidity of the layer model. A prototype combining a navigation interface with data stored in a LIS, has been implemented. This prototype, named GEOLIS, is used to illustrate concepts presented in this article. More precisely, we detail how the LIS notions of logics, object-centered information, querying, and navigation are instantiated for the purpose of GIS. In the next section, we introduce the data model used in GEOLIS. Section 3 is devoted to the navigation model. In Section 4, we describe the implementation of the prototype. And in the last two sections, we present results of experiments led on a real data set concerning the distribution of rodents in Sahelo-Sudanian Africa, and we compare our approach to existing works concerning information retrieval and logic applied to geographical data.

2. Data model

In GEOLIS data model, a geographical feature is represented by an object, and logics are used to describe and query objects. First of all, we introduce a formal definition of what we call a logic, and we show how logics are used to define the description language. Then, we introduce the notion of logical context and we illustrate it with the real dataset that we used in our experiments. Last, we detail the querying of the data model.

2.1. *Logic and description language*

A *logic* is a formal encapsulation of representation and reasoning. It is mainly composed of a language, whose elements are called *formulas*; and of a deduction relation, called *subsumption*, that tells us when a formula is more specific than another formula. Additional operations, such as conjunction, can be used.

Definition 1 (logic) *We define a* logic *as a partially ordered set* $\mathcal{L} = (L, \sqsubseteq, \sqcap)$*, where* L *is a set of logical formulas, and* $\sqsubseteq$ *is a* subsumption *relation between formulas, i.e., a partial ordering, and* $\sqcap$ *is a* conjunction *operation.*

In GEOLIS data model, each object represents a geographical feature and is described by a logical formula. This formula is a conjunction of logical properties derived from the semantic attributes and the spatial description of the feature (geometry and location). These properties are given a name and may be atomic or valued. They are the main elements of the description language (see Table 1).

Grammar of the description language

description	→	*description*
		`AND` *description*
	|	*descrProp*
descrProp	→	*name*`:`
	|	*name*`:`*value*

Grammar of the querying language

query	→	`(` *query* `)`
	|	`NOT` *query*
	|	*query* `OR` *query*
	|	*query* `AND` *query*
	|	`ALL`
	|	*queryProp*
queryProp	→	*name*`:`
	|	*name*`:`*formula*
formula	→	*value*
	|	*pattern*

Table 1. *Grammars of GEOLIS description and querying languages.*

Example: Here is an example of two formulas corresponding to a possible description of the French cities Rennes and Strasbourg in the GEOLIS data model:

- `name:“Rennes” AND point AND population:206 000 AND position:(351 869.83,6 789 643.91) AND data_provider:INSEE AND description:“administrative center of the French region Brittany”`
- `name:“Strasbourg” AND point AND population:263 941 AND position:(1 049 992.54,6 842 024.63) AND data_provider:INSEE AND description:“administrative center of the French region Alsace”`

In these descriptions, formulas are composed of logical properties combined with the conjunction operator `AND`. More in details, `point` is an atomic property about the geometric shape of the feature, `population` is an integer valued property, `name` and `description` are string valued properties and `position` is a coordinate valued property expressed in the coordinate system “Lambert 93”. Metadata about features such as data source or production date can also be expressed as properties, e.g. `data_provider:INSEE`.

As seen in the example, there are several domains of values depending on the type of properties. They can be simple (string, integer, float) or composite (coordinates), and have their own semantic and syntax. In LIS, each domain is defined as a specialized logic having its own language of formulas. This enables to use specific patterns in formulas, e.g. `population:>=100 000` which is more general than `population:206 000`.

The subsumption relation provides generalisation ordering between formulas: for instance,

– `population:206 000` $\sqsubseteq$ `population:206 000`,

– `population:263 941` $\sqsubseteq$ `population:>=200 000`,

– `population:>=200 000` $\sqsubseteq$ `population:>=100 000` $\sqsubseteq$ `population:`,

– `population:206 000 AND country:“France”` $\sqsubseteq$ `country:“France”`.

Left formulas are as precise or more precise than right ones and may describe fewer objects.

In fact reasoning on descriptions combines at the same time reasoning on the conjunction of logical properties and reasoning on the values of logical properties. More examples will be given in Section 2.3.

2.2. *Logical context and dataset for experimentation*

Once we have the logics to describe geographical objects, we need a structure to link each object to its description. This structure is called a *logical context*. A logical context is the encapsulation of a logic (which may be different from a context to another), and a set of objects accompanied with their logical description. This definition is from Logical Concept Analysis (LCA) on which LIS is based (Ferré *et al.*, 2004). A context is not static, but evolves through the addition, update, and removal of objects.

Definition 2 (context) *A* logical context *(or simply a* context*) is a triple* $(\mathcal{O}, \mathcal{L}, d)$*, where* $\mathcal{O}$ *is a finite set of objects,* $\mathcal{L}$ *is a logic, and* d *is a mapping from objects to logical formulas, i.e., denotes the logical description of objects.*

To illustrate the GEOLIS data model, we now indroduce the data set we used to make our experiments. It deals with the distribution of several species of rodents in Sahelo-Sudanian Africa. It is composed of one table where rows identify rodents and columns give descriptive information about these animals. This base, quite large (more than $20\,000$ individuals, potentialy described by 92 attributes), comes from the merging of several databases, the oldest data of which date back to 1903. Since 1980 the base is maintened by the French Institute for Research and Developpement (IRD) (L.Granjon, 2007). As rodent data come from local observations, this base is an imperfect sampling of the whole Sahelo-Sudanian stripe. It has been mainly designed to study the actual distribution of rodents, and to determine possible causes affecting this distribution.

In the *rodents context*, each object corresponds to a trapped rodent which is described by a conjunction of logical properties. These properties, expressed in French, inform about biometry (size aka `T_plus_C`, weight aka `Poids`, sexe, age), phylogeny (family aka `Famille`, genus aka `Genre`, species aka `Espece`), localisation (habitat, position where the animal was captured), and period of capture (aka `annee_capture`). The semantic diversity, the various domains of values available (string, integer, float, coordinates) and the simple geometry (point) of the features make this context an interesting candidate for first experiments of the GEOLIS data model. Table 2 shows a toy context representing a small part of the original rodent context.

2.3. *Logic and querying language*

Given a context, an important task is to compute the answers to a query. The answers are defined as the objects of the context whose description is subsumed by this query. The set of all answers to the query is called the *extent* of the query, which comes from LCA terminology.

Definition 3 (extent) *Let* $K = (\mathcal{O}, \mathcal{L}, d)$ *be some context, and* $q \in \mathcal{L}$ *be a logical formula representing a query. The* extent *of* q *in* K*, denoting the answers of the query* q *in the context* K*, is defined by*

$$ext(q) = \{o \in \mathcal{O} \mid d(o) \sqsubseteq q\}.$$

In order to avoid false negatives (missed answers), and false positives (wrong answers), it is important to ensure that the subsumption relation $\sqsubseteq$ is consistent and complete w.r.t. the semantics of formulas. For instance, $3 \sqsubseteq 0..5$ is correct, but $7 \sqsubseteq 0..5$ is incorrect because 7 does not belong to the interval $0..5$.

object o	*description* $d(o)$
$\mathtt{r}_1$	`Sexe:“F” AND Age:“Juv” AND Poids:38.0 AND annee_capture:2000 AND Famille:“Muridae” AND Genre:“Tatera” AND position:(-8.05,12.56)`
$\mathtt{r}_2$	`Sexe:“M” AND Age:“Ad” AND Poids:150.0 AND annee_capture:2001 AND Famille:“Muridae” AND Genre:“Arvicanthis” AND position:(-8.5,12.5)`
$\mathtt{r}_3$	`Sexe:“F” AND Age:“Ind” AND Poids:148.0 AND annee_capture:1999 AND Famille:“Sciuridae” AND Genre:“Xerus” AND position:(-8.41,11.96)`
$\mathtt{r}_4$	`Sexe:“M?” AND Age:“Ind” AND Poids:3.6 AND annee_capture:2001 AND Famille:“Muridae” AND Genre:“Mus” AND position:(-8.12,12) AND habitat:“parcelle de haricots”`
$\mathtt{r}_5$	`Sexe:“F” AND Age:“Ad” AND Poids:184.0 AND annee_capture:1998 AND Famille:“Muridae” AND Genre:“Arvicanthis” AND position:(-2.9,15.01) AND habitat:“brousse”`

Table 2. *A toy context. Descriptions are incomplete as they only include a part of the available properties. The age of rodents is declined in 3 classes: “Juv” for juveniles, “Ad” for adults, and “Ind” for unknown. Their position is expressed as (longitude, latitude) in decimal degrees.*

GEOLIS querying language extends the description language (see Table 1). It allows the use of disjunction and negation operators (resp. `OR` and `NOT`), parentheses in formulas, patterns from specialized logics allowing for building formulas subsuming a group of values, and the keyword `ALL` corresponding to the most general query, i.e. the query subsuming all object descriptions.

Specialized logics improve querying capabilities of GEOLIS. For instance a string logic enables to build patterns like `contains “administrative center”` which will cover the two cities of our previous example. In the toy context, an interval logic on the property `annee_capture` enables to define periods with patterns like `in 1999..2000`. In the same way, if attached to the weight property, it provides comparison operators like in the formula `Poids:>=100.0`. The empty pattern, like in query `Poids:` is the most general pattern of a logic. For instance, we have `Poids:148.0` $\sqsubseteq$ `Poids:>=100.0` $\sqsubseteq$ `Poids:`.

String logics and numeric logics enable to deal with semantic attributes. But semantic querying is a first step that has to be completed with spatial querying in order to retrieve the most information of geographical data. Spatial querying involves spatial logics. At the moment, GEOLIS supports a bounding box logic. In GIS, a bounding box corresponds to an axis-aligned rectangular envelope containing one or more features (see Figure 1). The most common way to define a bounding box is with the position of its lower left and upper right corner (resp. noted (x1,y1) and (x2,y2)). But as it is a rectangular area, it can also be considered as the product of two of its adjacent

sides (resp. written (x1..x2) and (y1..y2)). So a bounding box logic can be formed as the product of two interval logics on x and y coordinate values (see Section 4 for a description of the technology used to design this logic). This logic can be used to test if a ponctual feature belongs to a rectangular region on a map. For instance, in the rodent context, the trapping position of r_5 is subsumed by the query `position:in (-180.0..180.0,15.0..16.0)`, which represents the stripe delimited by latitude 15° N and 16° N. As we will see in next section, this is very useful when using a graphical interface, as rectangular zoom operations can be translated into bounding boxes. In the future, we plan to develop other logics to handle basic shapes including lines, polygons or disks.

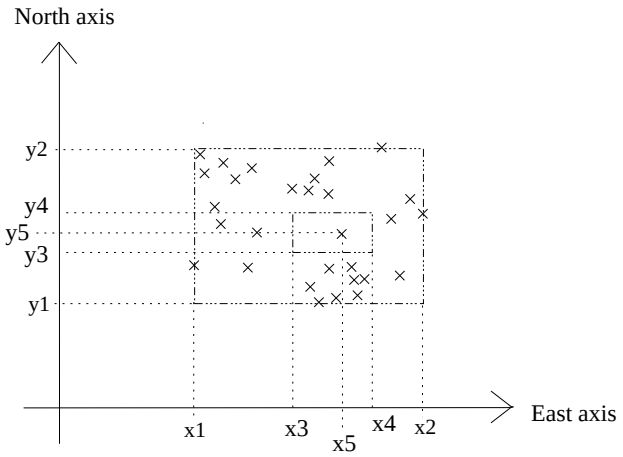

Figure 1. *Envelope ((x1..x2),(y1..y2)) represents the bounding box of all points of the region. It includes envelope ((x3..x4),(y3..y4)), which itself contains envelope (x3,y3) corresponding to the bouding box of only one point.*

When combining formulas in a query, the operator NOT is useful to exclude unwanted values like in `Famille:“Muridae” AND NOT Genre:“Arvicanthis”`. Naturally, the operator `OR` offers the disjunction of formulas like in the query q = `Age:“Ad” OR (Age:“Ind” AND Poids:>=100.0)` (See Table 3 for more examples of queries with their extents).

3. Navigation model

GEOLIS navigation model aims at suggesting to the user navigation links that lead him from a current place to a target place containing objects of interest. It is derived from the LIS navigation model and relies on the notion of query increments which is just introduced in the following. As GEOLIS deals with geographical features, we want it to render on a map the results of querying and navigation. That is why GEOLIS has to integrate a graphical interface. This interface and the corresponding navigation

q	$ext(q)$
`Sexe:“F”`	$\{r_1, r_3, r_5\}$
`Sexe:contains “M” AND Age:“Ad”`	$\{r_2\}$
`annee_capture:in 1995..2005 AND NOT annee_capture:2001`	$\{r_1, r_3, r_5\}$
`Famille:“Muridae” AND NOT Genre:“Arvicanthis”`	$\{r_1, r_4\}$
`Age:“Ad” OR (Age:“Ind” AND Poids:>=100.0)`	$\{r_2, r_3, r_5\}$
`Famille:“Muridae” AND (habitat:contains “parcelle” OR position:in ((-8.40..-8.60),(11.3..12.3)))`	$\{r_3, r_4\}$

Table 3. *Examples of queries with their extents about the toy context of Table 2.*

process are detailled in Section 3.2. Last, we discuss on the benefits of using GEOLIS for geographical information retrieval in comparison to traditionnal GIS.

3.1. *LIS navigation model*

The key contribution of LIS is to combine querying with a navigation that is automatically derived from the context. Querying is directly reaching a navigation place by giving a full query (possibly modified by hand from a previous query). As with file systems, we call *working query* the query that designates the current place. Navigating is following links from a working navigation place to other navigation places. Each link is represented by a query increment, i.e. a logical formula, and is automatically suggested by the system. These increments are defined as conjunctive refinements of the working query. If q_t denotes the working query at time t, $q_{t+1} = q_t \sqcap x_t$, where x_t denotes a query increment of q_t. LIS guaranty the *relevance* of the suggested query increments: when following a query increment, LIS ensure at the same time that the extent of the reached place has been reduced (w.r.t. the extent of the previous query), and that this extent is not empty. The need for this querying/navigation combination has already been recognized in other works (see Section 6). In LIS, every query reaches a navigation place that is characterized by a set of objects, the extent of the query. Reciprocally, every navigation place can be reached by one or several queries. For instance, the 2 queries $(q_1 \sqcap x_1) \sqcap x_2$ and $(q_1 \sqcap x_2) \sqcap x_1$ lead to the same navigation place, unlike in file systems (and other hierarchies), where 2 different paths lead to different directories.

Definition 4 (increments) *Let $K = (\mathcal{O}, \mathcal{L}, d)$ be a context, $X \subseteq \mathcal{L}$ be a fi nite subset of formulas, and $wq \in \mathcal{L}$ be a formula representing the working query, and denoting the working navigation place. The* query increments *from wq in K, denoting the way the query wq can be refi ned to reach relevant navigation places, are defi ned by*

$$incrs(wq) = \{x \in X \mid \emptyset \subset ext(wq \sqcap x) \subset ext(wq)\}.$$

To each increment is associated a number, called support*, that is equal to the cardinal of the extent $ext(wq \sqcap x)$, i.e. the number of objects that have this property. The subset X of formulas is the vocabulary used for increments. It is dependent on the chosen logic, and may be customized by users.*

For instance, in the toy context, consider $wq =$ `Poids:>=100.0`. Available query increments are all properties discriminating $ext(wq) = \{r_2, r_3, r_5\}$. They include `Poids:150`, `Poids:148`, `Poids:184`, `habitat:“brousse”`, `Age:“Ind”`, `Age:“Ad”` but not `Age:“Juv”` because $ext(wq \sqcap$ `Age:“Juv”`$) = \emptyset$. If we choose to refine wq with the increment `Genre:“Arvicanthis”`, then $wq =$ `Poids:>=100.0 AND Genre:“Arvicanthis”` and new query increments are generated. `Sexe:“M”`, `Sexe:“F”`, `Poids:150`, `Poids:184` are still available, but not `Poids:148`. No increments concerning the age, nor the family of rodents are proposed because they do not discriminate $ext(wq)$. Indeed, rodents of genus “Arvicanthis” always belong to family “Muridae”, this entailment is due to the phylogeny classification. The fact that all considered Arvicanthis rodents are adult depends on the context. This is called contextual entailment and may change when the context is updated.

3.2. *Navigation interface*

GEOLIS interface, which is shown in Figure 2, is composed of three main parts: the *navigation tree* placed on the left, the *map area* filling the center and the right, and the *working query box* at the bottom.

– The *working query box* displays the current query in the navigation. It indicates the query subsuming objects rendered in the map. The query box is editable, so that it is possible to enter manually a new query or to modify the current one.

– The *map area* is a composed component. A main map including fixed background layers (administrative boundaries, hydrography, isohyetal lines and satellite image) indicates by red points the position of rodents satisfying the current query. A legend details symbology of the main map and enables to specify which layers are visible. A keymap locates the boundaries of the main map on the Sahelian band. Last, standard map tools are also available: pan, zoom in/out and to full extent. The *map area* component comes almost unchanged from an existing interface (Mapserver community, 2007). It has been enhanced with a *logical zoom* tool, which enables to select rodents directly on the map by drawing a rectangle, i.e. a bounding box, enclosing them. This functionnality will be detailled in Section 3.3.

– The *navigation tree* is a visual representation of the partially ordered set of query increments. Query increments are properties shared by at least one rodent of *wq*. Each node of the tree represents a query increment which can be used to change wq (see Figure 2). A node can be expanded (resp. collapsed) to show (resp. to hide) its children, which represent more specific increments. The root of the tree is `ALL`, i.e., the most general formula. Nodes under the root correspond to general properties of the

taxonomy built over the dataset, e.g. `biometrie` whose children are `Age:`, `Poids:` or `Sexe:`. Then nodes represent pattern properties, e.g. `Poids:<=100`. Finally, value properties are the leaves of the tree, e.g. `Poids:70`. Each node of the tree is rendered with an icon, a label, and two numbers. The label is the formula representing the increment. The style of the label is also informative. Underlined orange labels correspond to formulas shared by all the rodents of wq, whereas blue labels indicate properties that discriminate them. The two numbers indicate a proportion: the count of rodents in wq that the increment leads to, i.e. the support, out of the count total of rodents sharing the formula. Two actions are possible in the tree: (1) collapsing or expanding a node by acting on the icon, (2) updating wq by selecting a label, as detailled in Section 3.3.

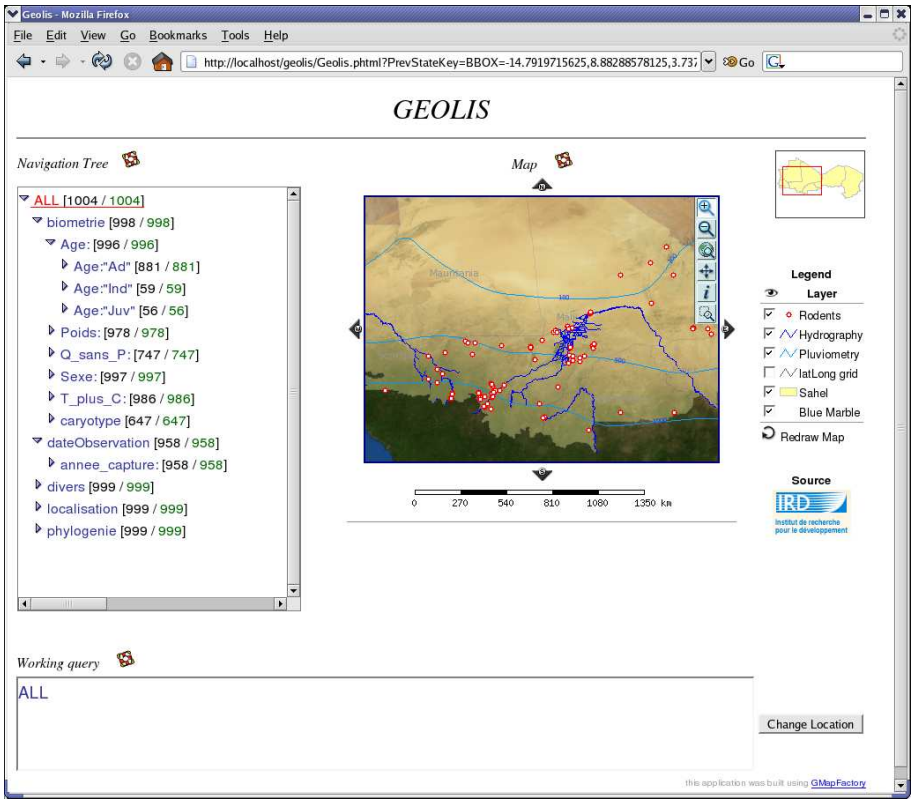

Figure 2. *The GEOLIS interface.*

3.3. *Navigation process*

During the navigation process, the interface is always maintained consistent. Each action on the *working query box*, the *navigation tree*, or the *map area* entails the

update of all the components. This is illustrated by the transition from Figure 2 to Figure 3. In Figure 2, no navigation is engaged, all objects are rendered on the map, wq is equal to the top formula `ALL`, and all navigation increments are available in the navigation tree. Let us suppose we are interested in the distribution of two families of rodents `“Murinae”` and `“Sciuridae”`. By editing the working query box and using the `OR` operator, we can restrict navigation to families of interest. In the same way, we can limit investigation to rodents captured since 2000, using the pattern `annee_capture:>=2000`. The selection of young rodents can be done by selecting the correponding increment `Age:“Juv”` in the navigation tree, or by manually updating the working query. The result of these navigation and querying steps appear in Figure 3: the map has been redrawn showing fewer points gathered mostly near water ressources in southern Malia or in the north of Burkina Faso border. At the same time, the new current query has been set, and the navigation tree updated: counts have been updated, increments that are no more relevant to the current query (e.g. `Age:“Ad”`, `Age:“Ind”`) have disappeared, and the formula `annee_capture:>=2000` has been inserted in the tree. Since a formula with a new pattern has been used in the working query, it appears in the navigation tree and can then serve as a query increment for the rest of the navigation.

We want now to focus on the set of rodents in southern Malia. With the *logical zoom* tool, we can draw on the map a rectangular shape enclosing the desired region (see Figure 3). This graphical selection entails the update of the current query, and consequently of the navigation tree and the map. The rectangular shape is translated into a formula based on `position` property, which is automatically added to the current query:

wq=`(Famille:"Muridae" OR Famille:"Sciuridae") AND annee_capture:>=2000 AND Age:"Juv" AND position:(-14.056..0.540,10.0585..13.393)`

The navigation tree is reduced, and now shows only properties and increments concerning rodents of the selected area.Furthermore, as expected, rodents in the north of the Burkina Faso border have disappeared from the map.

Notice that the logical zoom tool also plays the role of the information tool of standard GIS, which is used to query for the description of objects pointed on the map. In fact, when enclosing one feature with a logical zoom, its complete description can be read in the navigation tree, i.e. all, and no more, of the logical properties that qualify this feature are visible in the tree. When enclosing several features, orange underlined labels in the tree correspond to properties shared by all the selected features, whereas blue labels indicates discrimant properties, proper to fewer individuals. The common description of all selected rodents, which is not computed by traditionnal information tools, is here automatically provided by the navigation tree.

Once a logical zoom has been performed, it appears in the tree as an increment under the property `position`. Actually, a logical zoom is a graphical increment and relies on the same navigation mechanism than other query increments of the tree. The equivalent of the graphical selection can be obtained by entering the formula of the rectangle area in the working query box, or by simply selecting it in the navigation tree, if it has already been defined. However, directly acting with the map is faster,

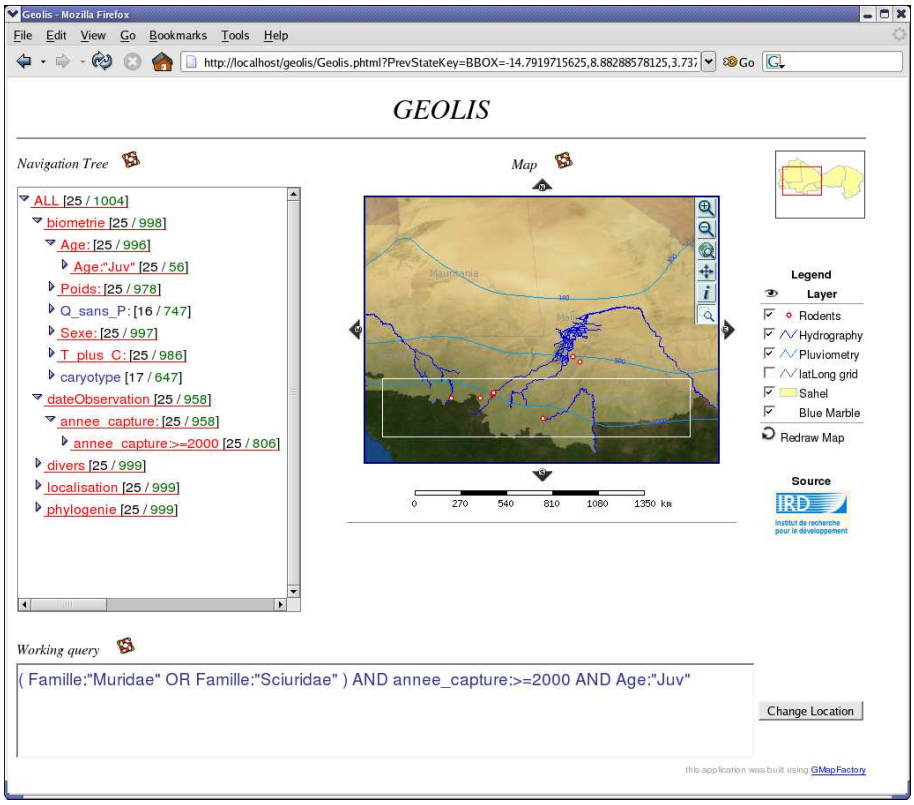

Figure 3. *Interaction between components. The white rectangle represents the area just selected by the logical zoom tool.*

enables to visually take into account the position of other geographical features, and, most of all, is more intuitive to users.

As logical zoom formulas are explicitly written in the query, they can be combined like other formulas using logical operators `AND`, `OR` and `NOT`. This way, complex areas can be defined from the combination of intersection, union and exclusion of rectangular regions. For instance, as shown in Figure 4, rodents trapped between the Niger River and the Black Volta river can be selected by combining enclosing and excluding logical zoom areas. After having drawn the 3 regions $R1$, $R2$ and $R3$ on the map, the desired area can be expressed by reorganizing the corresponding formulas in the working query :

$$wq = (\texttt{position:}R1\ \texttt{OR position:}R2)\ \texttt{AND NOT position:}R3 \quad .$$

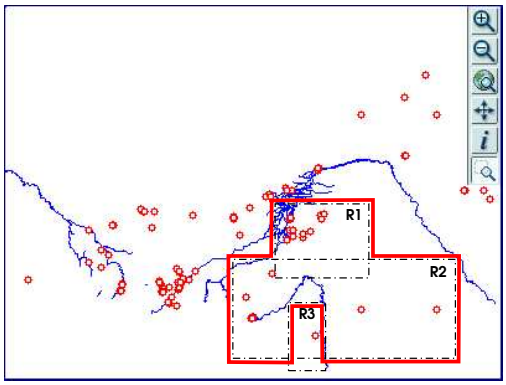

Figure 4. *Combining logical zoom rectangular areas (dashed outline) to defi ne a complex query area (plain outline).*

3.4. *Discussion about GEOLIS navigation functionnalities*

GEOLIS provides at the same time data navigation and querying functionnalites whereas traditionnal GIS, like databases, are rather designed for data querying. However, we can wonder how far it is possible to go with existing systems in order to provide similar navigation services. When working on data stored in an attribute table, several GIS, e.g. ArcGIS Desktop, SavGIS or GvSIG (ESRI, 2004; IVER *et al.*, 2007), or Relationnal Database Management Systems (RDBMS), e.g. Microsoft Access (MacDonald, 2007), already assist the user in the building of queries. For instance, they list all the available fields and their corresponding values for filling the SELECT and WHERE clauses. But this information does not take into account the extent of the queries and so does not always provide relevant increments for this query. Enhancing this kind of interface to help refining the query is totally conceivable. In the absence of patterns in value domains, an approximation of LIS operation $incrs(wq)$ would be in SQL language:

Foreach field $prop$ of the table $context$

`SELECT` $prop$`,count(*) AS c FROM` $context$ `WHERE` wq

`GROUP BY` $prop$ `HAVING c>0 and c<`c_{wq}

where c_{wq} corresponds to

`SELECT count(*) FROM` wq

and is computed once.

This implies $n + 1$ SQL queries at each navigation steps, with n denoting the number of fields of the table. However, such mechanism is artificial because it cannot be easily applied by GIS or RDBMS end-users. Moreover it cannot suggest pattern increments during the navigation, whereas GEOLIS does.

As already seen in previous examples, new patterns used with valued properties are introduced gradually in the tree, and enable to define classes of values just by navigation, when desired. This may be useful to specify weight groups or trapping pe-

riods. In the same way, each logical zoom introduces a new increment under property `position`. For instance, by defining horizontal stripes, we can look if the lattitude impacts the presence of high sized rodents of `"Gerbillus"` genus. Classes of values are not limited to numeric properties as it is the case in most desktop GIS. On the contrary, users can define patterns on string values, coordinates, and even dates. Moreover, thanks to the logics and the subsumption relation, classification is dynamic, i.e. when inserting new objects in the context, they automatically appear under suitable labels. This also allows for ranges of different sizes to be automatically ordered in the tree (e.g., for weight groups, `(1..10)`$\sqsubseteq$`(1..50)`$\sqsubseteq$`(1..100)`). The insertion of classes of value can also be automated using a script that navigates in each class at the first launch of GEOLIS.

GEOLIS navigation tree offers a synthetic vision puting together query increments. Supports of query increments provide a first intuition on the corresponding answer. Under each property node (e.g. `Age:`), the supports of child nodes (e.g. `Age:"Ad"`) provide the histogram of the distribution of values (see Figure 2). Furthermore, this representation is automatically derived from the LIS data organisation.

4. Implementation

The GEOLIS prototype results from the coupling of several technologies from LISFS, web mapping and web domains. First, we introduce LISFS, the implementation of LIS used in GEOLIS, and a way to build specialized logics. Then, we detail the components of the graphical interface and their interaction with LISFS.

4.1. *LISFS and specialized logics*

LISFS is a generic implementation of LIS, and is at the same time a genuine Linux file system (Padioleau *et al.*, 2003). In LISFS, files and file parts (lines) are objects, paths are queries, directories are navigation places, and subdirectories are the automatically computed query increments. Two kinds of plugins can be used in LISFS: logics and transducers. Logics define the kind of formulas that can be used in object description, and queries. Transducers allow to partially automate the description of objects, depending on the file format. For instance an MP3 transducer produces logical properties about the artist, title, etc., from the MP3 tags.

In LISFS, each property belongs to a logic that provides syntactic analysis for values and patterns comparison through deduction and pretty printing. However, designing a logic and ensuring its metalogic properties, i.e. the consistency and the completeness of its subsumption relation, requires logic expertise. This is why, in GEOLIS, we rely on LogFun, a toolbox of logic components which can be composed at a very high level (Ferré *et al.*, 2002). These components are called *logic functors*, and their composition results in a program defining a parser, a printer of formulas, and a subsumption tester. Futhermore, the consistency and the completeness of the

subsumption relation are checked during the composition; this automatically guarantees the good behaviour of the composed logic. Examples of logic functors are `Int` (integer values), `Float` (real values), `String` (string values and patterns such as "contains", "begins with"), `Interval(V)` (intervals over values in `V`), `Prod(L1,L2)` (product of 2 logics, so that formulas are couples of formulas). The bounding box logic, attached to the property `position`, has been formed as the product of two interval logics on real values: `Prod(Interval(Float),Interval(Float))`. In this case, the subsumption relation expresses inclusion of rectangular regions.

4.2. *GEOLIS architecture*

LISFS constitutes the kernel of GEOLIS, where the geographical data to be explored is stored, i.e. the rodent base in our experimentations. The GEOLIS graphical interface is a web interface. The navigation tree and the working query box have been designed using the server side language PHP. The map area is built with the widely used map generator UMN MapServer. Among the several geographical formats supported by MapServer, we chose to use the Geographical Markup Language (GML) proposed by the OpenGeospatial Consortium (Cox *et al.*, 2004). GML is an XML based format with public specifications. For our purpose, it has the advantages to gather all information in one file whose XML based structure may be rearranged w.r.t. to GML specifications. Furthermore, GML is supposed to become a standard for geographical data sharing.

We now detail the data flow of rodent information in GEOLIS (see Figure 5).

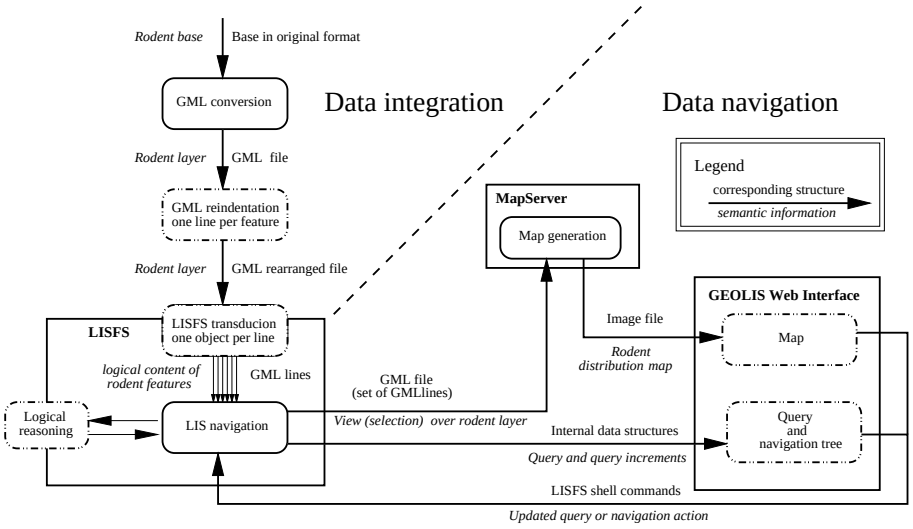

Figure 5. *Data flow of rodent information in GEOLIS. Dashed boxes correspond to functionalities specially designed for GEOLIS purpose.*

Two phases can be distinguished: data integration in LISFS, which is made once, and data navigation, which occurs at each step of logical navigation. In the data integration phase, the original rodent base, in MS Access format, is first translated in GML format using standard GIS conversion tools. Then, the GML file is reindented so that each geographical feature holds in one line. In the transdution process, i.e. when the GML file is parsed to extract descriptions of rodents, this reorganised structure enables to attach a LISFS object to each feature, as in LISFS objects are files or lines of files.

Once the GML file is mounted on LISFS, the navigation phase can start. At each navigation step, logical subsumption is used to determine which lines correspond to rodents satisfying the working query. In fact, this set of lines constitute a view over the whole GML file. This view, combined with complementary layers (hydrography, administrative limits, isohyetal lines and satellite image), is used by MapServer for the map generation. MapServer produces the general map, the keymap and the legend as images. At the same time, new query increments are computed by LISFS and transmitted to the web interface. Then, GEOLIS web interface generates the navigation tree and integrates the maps and the legend.

The GEOLIS prototype combines LISFS and logic functors, geographical data format and cartographic tools. These technologies were not designed to work together. However, their combination in GEOLIS did not require any modification. Much of the work have been to interface them, i.e. to determine the geographical format the most appropriate to LISFS integration, writting the corresponding transducer, building logics devoted to geographical data using LogFun, and designing the navigation tree interface.

5. Experimentations

During the development of the prototype, in order to validate navigation functionnalities, tests were first led on a subset of $1\,000$ rodents. However, for the experimentation phase, the whole base ($20\,585$ rodents with an average of 39 properties in each description) have been loaded in GEOLIS. Response times of navigation commands increased with the size of the context, but still allow human interaction (less than 10s on an experimentation machine, i.e. an Intel Pentium M 2Ghz with 1Go of RAM). That aspect mainly depends on LISFS, which is still under development and improvement.

First experiments in the rodent base highlighted several occurences of anomalous entries. These entries appear as properties with values out of the expected domain, e.g. `Sexe:“49”` instead of `“F”` or `“M”`, uncertain values, e.g. `Sexe:“?”` or even `Sexe:“M?”`, or synonymous values, e.g. `Sexe:“m”` and `Sexe:“M”`. These anomalies result from errors in data collecting and merging.

Initialy, spatial information in the rodent base was limited to the trapping position. So, to take into account the impact of other spatial factors on the distribution of rodents, some spatial relations, e.g. minimum distance from natural barriers (large

rivers) or closest upper and lower isohyetal lines, have been processed for each rodent using external GIS tools. Then they have been translated into semantic properties. This enables to provide spatial increments in data search. Furthermore, as the main map gives a visual representation of the location and concentration of rodents it rapidly suggested relevant spatial query increments during navigation.

As mentioned previously, the rodent base comes from an imperfect sampling. This has been observed in the navigation. For instance, by just expanding the node `pays` (aka country) in the navigation tree and observing the count of rodents associated with each value, we noticed that half of information in the base comes from Senegal which clearly appears on the map to represent a small part of the studied area.

Then, when expanding node `annee_capture` (trapping year), we could see jointly, in two branchs of the tree, the distribution of rodents by year or place of trapping. Refining the working query with query increments under property `pays` showed in one navigation step the evolution of this distribution per year when restricted to a particular country. This way, we noticed that in Chad, recent data have only been collected in year 2000, whereas in Mali and Senegal, data are available at least every two years.

Having knowledge about data origin could enable to balance future results concerning rodents distribution. So we decided, as a first step, to study the sampling strategies in the database. For instance, we looked for connections between rodents trapped states (alive, dead), places and periods of capture. As explained previously, we can restrict navigation to rodents trapped alive, and visualize, at the same time, properties `annee_capture` and `habitat`. Selecting a particular habitat, e.g. `savane` (savanna), and looking at property `annee_capture` show for each year, how many rodents were trapped alive in the savane. On the opposite, distribution of trapping places concerning a particular year could be observed by looking at `habitat` and selecting `annee_capture`. This shows that GEOLIS is appropriate to quickly check distribution hypotheses implying several criteria. In this sense, LIS play a similar role to OLAP processing (Chaudhuri *et al.*, 1997, On-Line Analytical Process).

6. Related works

The need for combining navigation and querying in information retrieval has already been underlined in other works (Godin *et al.*, 1993; Chiaramella, 1997). Advanced file systems coupling hierarchical navigation and boolean querying have also been proposed. For instance, Semantic File System (Gifford *et al.*, 1991, SFS) enables, in addition to traditionnal directories organisation, to define virtual directories as queries built from intrinsic properties of files, e.g. file type or date of last modification. But in SFS, navigation cannot be used after a querying step. In Hierarchy and Content (Gopal *et al.*, 1999, HAC), another attempt to mix navigation and querying at the file system level, each directory is expressed as a query that defines its content. However, the navigation is limited to the sub-directories that have been manually cre-

ated in a directory/query. Futhermore, the user can freely move files into directories, even if they do not satisfy the corresponding query. So nothing guaranties that the navigation structure is kept consistent. On the contrary, LIS allow to freely alternate navigation and querying, and rely on a navigation structure automatically derived from the data, which therefore guaranties its consistency.

The use of logical formalisms in spatial representation has been widely explored in the last two decades. Especially, several approaches have been proposed to model topological relations (Asher *et al.*, 1995; Cohn, 1997). Modal logics have also been used to define spatial logics, including for instance the notion of proximity (Lemon, 1996). However these approaches are rather qualitative and often not directly useable with real world data. Description logics also allow to represent domain knowledge, and have led to some encouraging attempts to describe the geographical domain. For instance, they have inspired spatial representations used in the LOLA system, wich is devoted to the recognition and the classification of spatial structures from satellite images (Le Ber *et al.*, 2002). The VISCO system (Wessel *et al.*, 2000) can be more closely compared to GEOLIS as the main aim of the two systems is geographical information retrieval. In the VISCO system description logics are used to query a spatial database in a visual way. VISCO enables to represent and to reason on topoligical relations between geographical features. This lacks GEOLIS at the present time, since relations and especially spatial relations cannot be expressed in the current version of the system. VISCO integrates querying capabities, and can also assist the user with query completion. Completion represent a form of navigation, however in VISCO, the proposed completion comes from terminological default reasoning (Wessel *et al.*, 2000), and may produce queries with an empty answer. Whereas in GEOLIS, query increments are always relevant w.r.t. the objects satisfying the current query.

GEOLIS, and more generally LIS, combine a quantitative approach as data are described by expressive logics on concrete domains, and a qualitative approach, derived from the former, through conceptual structures and logical reasoning. Furthermore, GEOLIS provides a geovisualization synchronized with a navigation accross the different dimensions (properties) of a geographical dataset and the possibility to rapidly group and visualize data values. These functionnalities, proper to SOLAP tools (Rivest *et al.*, 2005, Spatial OLAP), offer an iterative and interactive exploration of a geographical dataset.

7. Conclusion and Prospects

GEOLIS is a framework where Logical Information System principles have been applied to geographical data. It is important to note that the proposed method for managing GIS data is compatible with existing data, and with existing cartographic interfaces. Thanks to the transducers it handles data as it is, and thanks to LISFS being a file system, it is easy to make a standard interface operate on a logical context. The interface needs only be extended to handle LISFS navigation and querying.

The GEOLIS experiments have been conducted with real data, accumulated over years by different teams that were completely unrelated with the GEOLIS group. So, the dataset was not formatted at all to fit the GEOLIS data model. But LISFS naturally offers functionnalities that facilitate the data analysis process: the transducers and the taxonomy made possible to quickly integrate data and organize properties and the navigation links having low support enabled to identify anomalous entries. A main contribution of GEOLIS is to facilitate the exploration of data, and to quickly check experts hypotheses.

In the future, we plan to work on spatial logics to improve expressiveness and querying capabilities of GEOLIS. Data representation should include derived geometrical properties, such as area and length. Navigation should integrate graphical query increments, and the possibility of automatic zooming on the region of interest. GEOLIS querying language is not yet as expressive as traditionnal GIS SQL-based languages (for instance, aggregates are not possible yet). However, we believe our logics of values and patterns for representing various kinds of properties (e.g., coordinates or geometries) are more intuitive to users. This idea follows principles of naive geography, which aims at designing GIS "that follows human intuition" (Egenhofer *et al.*, 1995). Furthermore, LISFS is in permament evolution. In particular, relations (Ferré *et al.*, 2005) will be soon implemented, including spatial relations in the future. This will enable to express spatial relations between features, such as distances, topological relations and to look for spatial organisation patterns. LISFS should also integrate data-mining operations for finding association rules. This will offer these data-mining operations to the GIS domain almost immediately, and will make emergent relationships between properties visible. In the rodent experiments, these improvements would enable, for instance, to look for spatial barriers in the distribution of rodents.

Acknowledgements

The authors would like to thank M.Laurent Granjon and M. Jean Marc Duplantier from IRD (CBGP UR 22 Montpellier) for their active contribution in the building of the rodent database.

8. References

Asher N., Vieu L., "Toward a Geometry for Common Sense: A Semantics and a Complete Axiomatization for Mereotopology", *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1995.

Bruns T., Egenhofer M., "User Interfaces for map Algebra", *Journal of the Urban and Regional Information Systems Association*, vol. 9, n° 1, p. 44-54, 1997.

Chaudhuri S., Dayal U., "An overview of data warehousing and OLAP technology", *SIGMOD Rec.*, vol. 26, n° 1, p. 65-74, 1997.

Chiaramella Y., " Browsing and Querying: Two Complementary Approaches for Multimedia Information Retrieval.", *HIM*, p. 9-26, 1997.

Cohn A. G., "Qualitative Spatial Representation and Reasoning Techniques", *KI '97: Proceedings of the 21st Annual German Conference on Artificial Intelligence*, 1997.

Cox S., Daisey P., Lake R., Portele C., Whiteside A., *Geography Markup Language (GML) Encoding Specification*, Open Geospatial Consortium (OGC). 2004.

Egenhofer M. J., Mark D. M., "Naive Geography", *COSIT'95*, 1995.

ESRI (ed.), *Using ArcMap: ArcGIS 9*, ESRI Press, 2004.

Ferré S., Ridoux O., " A Framework for Developing Embeddable Customized Logics", *in* A. Pettorossi (ed.), *Int. Work. Logic-based Program Synthesis and Transformation*, LNCS 2372, Springer, p. 191-215, 2002.

Ferré S., Ridoux O., "An Introduction to Logical Information Systems", *Information Processing & Management*, 2004.

Ferré S., Ridoux O., Sigonneau B., "Arbitrary Relations in Formal Concept Analysis and Logical Information Systems", *ICCS*, LNCS 3596, Springer, p. 166-180, 2005.

Gifford D. K., Jouvelot P., Sheldon M. A., James W. O'Toole J., " Semantic file systems", *SIGOPS Oper. Syst. Rev.*, vol. 25, n° 5, p. 16-25, 1991.

Godin R., Missaoui R., April A., "Experimental Comparison of Navigation in a Galois Lattice with Conventional Information Retrieval Methods.", *International Journal of Man-Machine Studies*, vol. 38, n° 5, p. 747-767, 1993.

Gopal B., Manber U., " Integrating content-based access mechanisms with hierarchical file systems", *OSDI '99: Proceedings of the third symposium on Operating systems design and implementation*, USENIX Association, Berkeley, CA, USA, p. 265-278, 1999.

IVER, Conselleria Valenciana d'Infraestructures i Transport, *GvSIG Official web site*. 2007, http://www.gvsig.gva.es/index.php?L=2.

Laurini R., Thompson D., *Fundamentals of Spatial Information Systems*, Academic Press Limited, 1992.

Le Ber F., Napoli A., "Object-based Representation and Classification of Spatial Structures and Relations", *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'02)*, 2002.

Lemon O. J., "Semantical Foundations of Spatial Logics", *in* L. C. Aiello, J. Doyle, S. Shapiro (eds), *KR'96: Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, San Francisco, California, p. 212-219, 1996.

L.Granjon, Inventaire et caractérisation des espèces de rongeurs sahélo-soudaniens, Technical report, IRD, 2007. http://www.mali.ird.fr/activites/inventaire.htm.

MacDonald M., *Access 2007 for starters: The Missing Manual*, O'Reilly, 2007.

Mapserver community, *MapServer official web site*. 2007, http://www.mapserver.gis.umn.edu.

Padioleau Y., Ridoux O., "A Logic File System", *Usenix Annual Technical Conference*, 2003.

Rivest S., Bédard Y., Proulx M.-J., Nadeau M., Hubert F., Pastor J., "SOLAP: Merging Business Intelligence with Geospatial Technology for Interactive Spatio-Temporal Exploration and Analysis of Data", *ISPRS*, vol. 60, n° 1, p. 17-33, 2005.

Wessel M., Haarslev V., Möller R., " Visual Spatial Query Languages: A Semantics Using Description Logic", *Diagrammatic Representation and Reasoning*, Springer, 2000.

ANNEXE POUR LE SERVICE FABRICATION
A FOURNIR PAR LES AUTEURS AVEC UN EXEMPLAIRE PAPIER
DE LEUR ARTICLE ET LE COPYRIGHT SIGNE PAR COURRIER
LE FICHIER PDF CORRESPONDANT SERA ENVOYE PAR E-MAIL

1. ARTICLE POUR LA REVUE :

Revue internationale de Géomatique. Volume X – n°x/200X

2. AUTEURS :

O. Bedel[*1] *— S. Ferré*[*] *— O. Ridoux*[*] *— E. Quesseveur*[**]

3. TITRE DE L'ARTICLE :

GEOLIS : A Logical Information System for Geographical Data

4. TITRE ABRÉGÉ POUR LE HAUT DE PAGE MOINS DE 40 SIGNES :

GEOLIS

5. DATE DE CETTE VERSION :

June 23, 2007

6. COORDONNÉES DES AUTEURS :

– adresse postale :

[*] Univ. Rennes 1/IRISA, Equipe LIS
Campus de Beaulieu
35042 Rennes Cedex FRANCE
{prénom.nom}@irisa.fr

[**] RESO, UMR CNRS ESO 6590
Univ. Rennes 2, Campus Villejean
35 043 Rennes Cedex FRANCE
{prénom.nom}@uhb.fr

– téléphone : 02 99 84 73 29
– télécopie : 02 99 84 71 71
– e-mail : olivier.bedel@irisa.fr

7. LOGICIEL UTILISÉ POUR LA PRÉPARATION DE CET ARTICLE :

LaTeX, avec le fichier de style `article-hermes.cls`,
version 1.23 du 17/11/2005.

8. FORMULAIRE DE COPYRIGHT :

Retourner le formulaire de copyright signé par les auteurs, téléchargé sur :
`http://www.revuesonline.com`

SERVICE ÉDITORIAL – HERMES-LAVOISIER
14 rue de Provigny, F-94236 Cachan cedex
Tél. : 01-47-40-67-67
E-mail : revues@lavoisier.fr
Serveur web : http://www.revuesonline.com