

---

# GEOLIS : A Logical Information System for Geographical Data

O. Bedel\*<sup>1</sup> — S. Ferré\* — O. Ridoux\* — E. Quesseveur\*\*

\* Univ. Rennes 1/IRISA, Projet Lande  
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 prevailing structure in geomatics for handling geographical information. However, the layer model is rigid: it implies partitioning 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 thinner level since it is centered on the geographical feature. LIS does not rely on a hierarchical organisation of information, and enable to tightly combine querying and navigation in a same search. In this article, we present a work in progress about the use of LIS model 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 with 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) offrent un nouveau paradigme pour l'organisation et la recherche d'information. Avec les SIL, nous proposons un modèle d'organisation plus flexible des données géographiques vectorielles, centré sur l'entité géographique. Les SIL n'imposent aucune structuration hiérarchique de l'information et permettent de combiner étroitement, dans une même recherche, interrogation et navigation. Dans cet article, nous présentons nos travaux sur l'utilisation des SIL dans le cadre géographique. Nous détaillons un modèle de données et le modèle d'interrogation et de navigation associé. Le prototype GEOLIS nous permet de présenter ces modèles appliqués 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 prevailing model for handling information in Geographical Information Systems (GIS). This structure gathers geographical data under a common theme e.g., soils, roads, water [LAU 92], 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 [BRU 97]. However, this structure is rigid as it implies partitioning geographic information in predefined categories, and usually having the same description schema for all the elements of a layer (pixels for raster data or features for vectorial data). 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). It uses logic in a uniform way to describe data, query it, to navigate through it and to update it. The LIS model has been implemented as a filesystem; this permits to integrate this new methodology to existing applications.

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 to get free from 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. In the next section, we introduce general notions and definitions relative to the LIS model. Sections 3 and 4 detail the data model and the navigation model used in GEOLIS. In section 5, we describe the implementation of the prototype. Last section presents results of experiments led on a real data set concerning the distribution of rodents in Sahelo-Sudanian Africa.

## 2. From LIS to GEOLIS

Logical Information Systems (LIS) offer a new paradigm for information management and retrieval. This paradigm is characterized by the following principles [FER 04]:

- information is centered on the objects of interest, i.e., on the entities one wish 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.

We detail in the following the LIS notions of logic, object-centered information, querying, and navigation. These notions are quite general, and are instantiated for the purpose of GIS in the next sections.

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, we call *subsumption*, that tells us when a formula can be deduced from 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.

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 [FER 04]. 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.

Given a context, an important task is to compute the answers to a query. These answers are defined as the objects whose description is subsumed by the 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  $[0, 5]$ .

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). 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, and ensure that the extent of the reached place is not empty. The need for this querying/navigation combination has already been recognized in other works, but in most proposed solutions, either navigation cannot be used after a querying step, or the navigation structure is rigid as in file systems, or the navigation structure is not kept consistent with object descriptions when they change. 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. As with file systems, we call *working query*, the query that designates the current place.

**Definition 4 (increments)** *Let  $K = (\mathcal{O}, \mathcal{L}, d)$  be a context,  $X \subseteq \mathcal{L}$  be a finite 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 refined to reach relevant navigation places, are defined by*

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

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

LIS are generic in that the logic is not fixed *a priori*, but can be chosen to match closely the needs of a particular application. LISFS is a generic implementation of LIS, and is at the same time a genuine Linux file system [PAD 03]. 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 the next sections, we show how LIS and LISFS are used to build a Geographical LIS, GEOLIS. The logic is first defined in Section 3, then logical querying and navigation are illustrated on a real dataset in Section 4.

### 3. Data model

In GEOLIS data model, a geographical feature is represented by an object, and logics are used to describe and query objects. We first define the description and querying languages. Then we present the real dataset that has been used to test the model.

<i>Grammar of the description language</i>		<i>Grammar of the querying language</i>	
<i>description</i>	→ <i>description</i>   <b>AND</b> <i>description</i>	<i>query</i>	→ <b>NOT</b> <i>query</i>   <i>query</i> <b>OR</b> <i>query</i>   <i>query</i> <b>AND</b> <i>query</i>
<i>descrProp</i>	→ <i>name</i> :   <i>name</i> : <i>value</i>		<b>ALL</b>   <i>queryProp</i>
		<i>queryProp</i>	→ <i>name</i> :   <i>name</i> : <i>formula</i>
		<i>formula</i>	→ <i>value</i>   <i>pattern</i>

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

### 3.1. Description language

Each object represents a geographical feature and is described by a conjunction of properties derived from the semantic attributes and a 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).

There are several domains of values depending on the type of properties. They can be simple (string, integer, float) or composite (coordinates). Each domain is defined as a specialized logic (see Section 2) having its own language of formulas.

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

- name:"Rennes" AND point AND population:206000 AND position:(351869.83,6789643.91) AND data\_provider:INSEE AND description:"administrative center of the French region Brittany"
- name:"Strasbourg" AND point AND population:263941 AND position:(1049992.54,6842024.63) AND data\_provider:INSEE AND description:"administrative center of the French region Alsace"

In this description, *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.

### 3.2. Querying language

For the purpose of information retrieval, a querying language extends the description language (see Table 1) with disjunction and negation operators (resp. **OR** and **NOT**), 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.

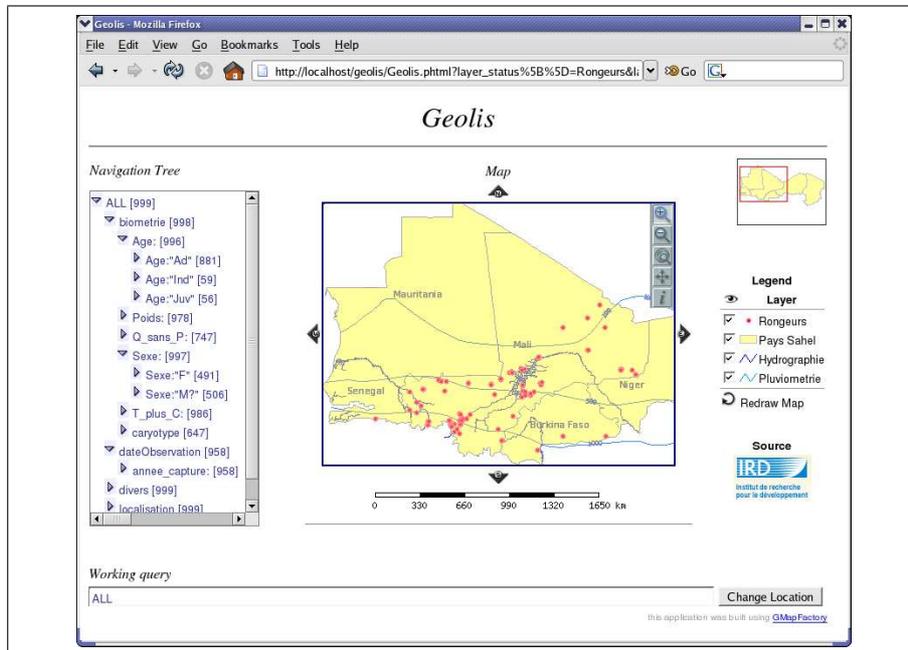
Several specialized logics may be used in GEOLIS to improve querying capabilities. For instance a string logic enables to build patterns like contains “administrative center” which will cover the two cities of our example. An interval logic on integer values provides patterns useable in queries like `population:in 100000..300000` or `population:>=200000`. The empty pattern, like in query `population:` is the most general pattern of a logic. For instance, we have `population:263941`  $\sqsubseteq$  `population:>=200000`  $\sqsubseteq$  `population:`. A coordinate logic can be defined as the combination of interval logics on real values. It enables to build patterns corresponding to axis-aligned rectangular areas. For instance, Strasbourg position (1049992.54,6842024.6) is covered by the query (1000000..1100000,6700000..6900000).

### 3.3. Dataset for experimentation

As an illustration of the GEOLIS data model, we now introduce 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, potentially described by 92 attributes), comes from the merging of several databases provided and maintained by the French Institute for Research and Developpement (IRD) since 1980 [IRD]. As rodents data come from local observations, this base is an imperfect sampling of the whole Soudano Sahelian stripe. It has been mainly designed to study the actual distribution of rodents, and to determine possible causes affecting this distribution. Therefore, the description of each rodent mainly contains information about biometry (size, weight, sexe, age), phylogeny (family, genus, specy), localisation (position where the animal was captured, habitat), and period of capture. This semantic diversity, the various domains of values available (string, integer, float, coordinates) and the simple geometry (point) of the features make this base an interesting candidate for first experiments of the GEOLIS data model.

## 4. Navigation model

GEOLIS navigation model is derived from the LIS navigation model. It aims at suggesting to the user navigation links that lead him from a current place to a target place containing objects of interest. The model relies on three main elements, introduced in Section 2: the working query  $wq$  (i.e. the current place), the query increments (i.e. the navigation links between places), and the extents (i.e. the objects selected by  $wq$  in the current place). As GEOLIS deals with geographical features, we want it to render maps, as a result of the visual representation of the extent in a cartographic space. So, GEOLIS has to integrate a graphical interface, which is shown in Figure 1.



**Figure 1.** *The GEOLIS interface.*

#### 4.1. *Navigation interface*

The interface 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.

- The *map area* is a composed component. A main map including fixed background layers (administrative boundaries, hydrography and isohyetal lines) 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 on the main map. A keymap locates the boundaries of the main map on the Sahelian band. Last, standard map tools are available: pan, rectangular zoom in, zoom out and zoom to full extents. This component comes almost unchanged from an existing interface [MAP].

- The *navigation tree* is a visual representation of the partially ordered set of query increments. Each node of the tree represents a query increment which can be used to change *wq* (see Figure 1). 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 reminding the increment, and a number indicating the count of objects, i.e. the support, in the place pointed by the increment. Two actions are possible in the tree. First, collapsing or expanding a node by acting on the icon. Then, updating *wq* by selecting a label. The navigation tree shows in one shot many subqueries of the working query and the size of the corresponding answers. Furthermore, 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 1).

#### 4.2. Navigation process

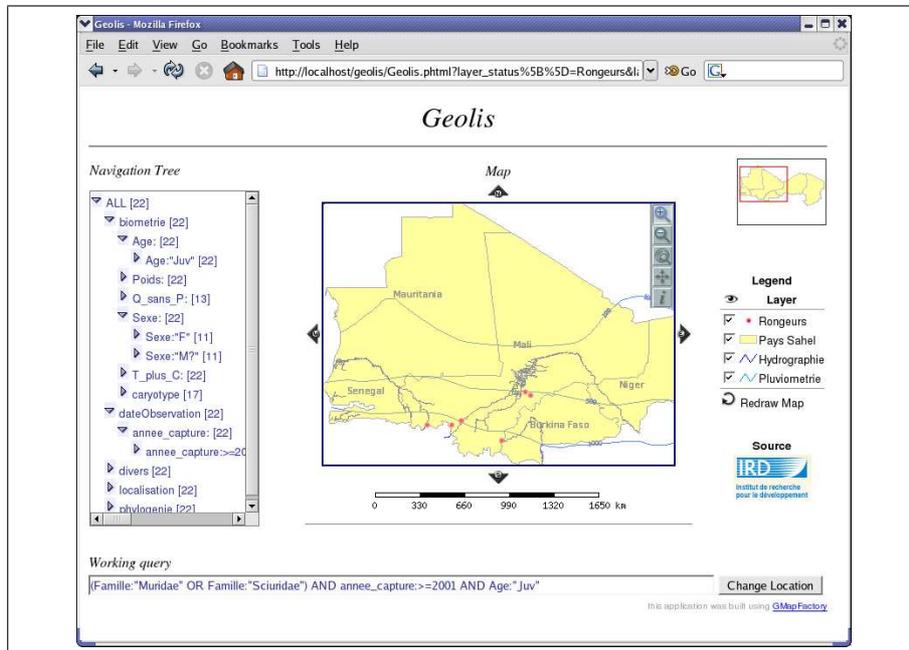
During the navigation process, the interface is always maintained coherent. Each action on the *working query box* or the *navigation tree* entails the update of all the components. This is illustrated by the transition from Figure 1 to Figure 2. In Figure 1, no navigation is engaged, all objects are 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 2001, using the pattern `annee_capture:>=2001`. The selection of young rodents can be done by selecting the corresponding 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 2: the map has been redrawn showing fewer points gathered mostly near water resources, the new current query has been set, and the navigation tree updated, the cardinalities have been updated, and increments no more relevant to the current query (e.g. Age:"Ad", Age:"Ind") have disappeared.

The navigation tree offers a synthetic vision of query increments which makes navigation and querying easier and faster. Supports of query increments provide a first intuition on the corresponding answer. Furthermore, this representation is automatically derived from LIS data organisation. Within traditional GIS, maintaining such a navigation tree would imply executing at each navigation step many queries.

It is also planned that map area become an entry point for logical navigation. A logical zoom tool would enable to define spatial query increments represented by rectangular region drawn on the map. These spatial increments would refine the position property.

### 5. Implementation

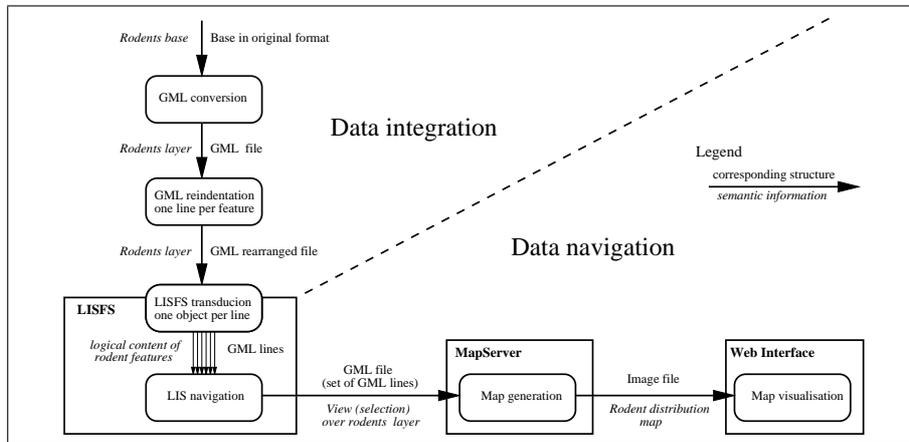
The GEOLIS prototype results from the coupling of several technologies from LISFS, web mapping and web domains. LISFS (see Section 2) constitutes the ker-



**Figure 2.** Interaction between components.

nel of GEOLIS, where the geographical data to be explored is stored, i.e. the rodents 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. Between the several geographical formats supported by MapServer, we chose to use the Geographical Markup Language (GML) proposed by the Open-Geospatial Consortium [COX 04]. 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 rodents information in GEOLIS (see Figure 3). 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 rodents 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 transduction 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, only lines corresponding to rodents satisfying the working query



**Figure 3.** Data flow of rodents information in GEOLIS.

are visible. In fact, this set of lines constitute a view over the whole GML file. This view, combined with complementary layers (hydrography, administrative limits and isohyetal lines), is used by MapServer for the map generation. MapServer produces the general map, the keymap and the legend as images, which are incorporated in the web interface.

The GEOLIS prototype combines LISFS, 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, writing the corresponding transducer, building logics devoted to geographical data and designing the navigation tree interface.

## 6. Experimentations

During the development of the prototype, in order to validate navigation functionalities, 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 rodents base highlighted several occurrences 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.

Initially, spatial information in the rodents 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 rodents using external GIS tools. Then they have been translated into semantic properties. This enables to provide pseudo-spatial increments in data search. Furthermore, as the main map gives a visual representation of the location and concentration of rodents it could rapidly suggest relevant spatial query increments during navigation.

As mentioned previously, the rodents base comes from an imperfect sampling. This has been observed in the navigation. For instance, by just expanding the node `pays(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 branches 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.

## 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 functionalities 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, and enable to express spatial relations between features, such as distances, topological relations, distributions and spatial organisation patterns. 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 traditional 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” [EGE 95]. Furthermore, LISFS is in permanent evolution. It will soon 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 rodents 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 rodents database.

#### 8. References

- [BRU 97] BRUNS T., EGENHOFER M., “User Interfaces for map Algebra”, *Journal of the Urban and Regional Information Systems Association*, vol. 9, num. 1, 1997, p. 44-54.
- [COX 04] COX S., DAISEY P., LAKE R., PORTELE C., WHITESIDE A., “Geography Markup Language (GML) Encoding Specification”, Open Geospatial Consortium (OGC), 2004.
- [EGE 95] EGENHOFER M. J., MARK D. M., “Naive Geography”, *COSIT'95*, 1995.
- [FER 04] FERRÉ S., RIDOUX O., “An Introduction to Logical Information Systems”, *Information Processing & Management*, vol. 40, num. 3, 2004, p. 383-419.
- [IRD] “Inventaire et caractérisation des espèces de rongeurs sahélo-soudaniens”, Institut de recherche pour le développement (IRD), <http://www.mali.ird.fr/activites/inventaire.htm>.
- [LAU 92] LAURINI R., THOMPSON D., *Fundamentals of Spatial Information Systems*, Academic Press Limited, 1992.
- [MAP] “MapServer”, Official Web Site, <http://www.mapserver.gis.umn.edu>.
- [PAD 03] PADIOLEAU Y., RIDOUX O., “A Logic File System”, *Usenix Annual Technical Conference*, 2003.