

Resolution of Ambiguities in Query Interpretation for Geographical Pictorial Query Languages

Fernando Ferri¹, Maurizio Rafanelli²

¹ IRPPS-CNR, Roma, Italy

² IASI-CNR, Roma, Italy

The main problem of visual query languages for geographical data concerns query ambiguity. A query can have different visual representations, which in turn can have different interpretations.

Increasing the number of query objects increases its ambiguity. This derives from the fact that a query can lead to multiple interpretations for both the system and user. The user's actions may not represent his intentions, leading the system to an incorrect interpretation. So the user cannot express his exact query and different queries must thus be formulated to achieve his goals.

This paper proposes an approach that allows the user to represent only desired constraints and avoid undesired constraints in visual query representation.

Keywords: query language, GIS, ambiguity.

1. Introduction

In visual query languages, a query can lead to multiple interpretations for the system and user. One of the main reasons is that a unique working space is often used to represent and express different kinds of information. Another is due to the different approaches of the user in formulating his query and the system in its analysis.

To date, two main approaches have been developed in designing visual query languages: in the first, the user draws his query directly on the screen, using the blackboard metaphor. Examples of this approach are Sketch [1] and Spatial-Query-By-Sketch [2]. The user is free to draw shapes or symbols on the screen by hand to express an operator or an object.

In the second approach, the user uses commands such as buttons to display metaphorical patterns (icons or symbols) in a command user interface. Important examples of this approach are the Cigales language [3, 4] and the Lvis language [5, 6].

Cigales allows the user to draw a query. It is based on expressing a query by drawing the pattern corresponding to the user's desired result. Graphical forms and icons that conceptualize the operators are predefined.

Lvis is an extension of Cigales. The most relevant difference is its definition of new operators, because it considers both spatial and temporal properties of the objects forming the query. Here too, the main limitation is different interpretations of the same query, that is, the system's inability to provide a unique interpretation of a given visual query. The more query objects, the higher the ambiguity.

In [7] the authors confront this problem in visual GIS query languages and propose a taxonomy based on user actions and system materialization (the different images it can materialize), to distinguish ambiguities in visual GIS query languages and ways to resolve them, and a model to solve a particular case of ambiguity. The proposed system, an enlargement of Lvis, establishes a dialog with the user and whenever an ambiguity occurs, it shows all available configurations and requests a choice. The author concludes that the strategy to avoid ambiguities in most visual geographic query languages is to define not fully visual, but hybrid languages,

including a textual part and offering a grammar with low expressive power.

In Section 2 of this paper we give a brief overview of ambiguity in visual query languages for geographical data. In Section 3 we propose the Geographical Pictorial Query Language (GeoPQL) and discuss its algebra in a geographical data context. In Section 4 we see how GeoPQL resolves any ambiguities arising in graphical query representation. In Section 5 we show how a pictorial query is resolved by the implemented system. Finally, Section 6 gives our conclusions.

2. Visual Representations and Query Ambiguities

Visual languages often use icons and metaphors to model spatial objects, express spatial relationships between objects and then formulate queries. They offer an intuitive and incremental view of spatial queries, but often have little expressive power, fairly ineffective query execution and may offer different interpretations of the same query (the ambiguity problem).

For example, suppose the user wants to formulate the following query: “Find all the regions which are *passed-through* by a river and *overlap* a forest”. In this query the user is not interested in the relationship between the river and the forest. However, when the user draws a shape representing a region and another representing a river, he cannot avoid representing a spatial relationship between them, so any representation considering a specific relationship between the two features can be considered as valid. Different pictorial queries can thus represent the previous query in natural language. In particular, in Figure 1-a the forest and the river are “Disjoint”, in 1-b the river touches the forest and in 1-c the river passes through the forest. It is also possible to consider other visual queries representing the same query. When a parser processes the queries in Figure 1, it should consider all constraints represented in the pictorial representation. For this reason, the three representations should be interpreted as three different queries, each having a different meaning from that of the original in natural language.

Different approaches have been proposed to remove undesired constraints.

In one, the user can explicitly remove some of the constraints between drawn icons or impose an a priori restrictive interpretation using the foreground/background concept. When he draws a new icon, the parser considers the relationships between itself and other icons in the foreground, and removes the relationships between itself and the others in the background.

The relationships considered thus depend on the previously drawn features. The relationship between a newly drawn ϕ_i and a previous feature ϕ_j is considered if ϕ_j is in Foreground state. It is not considered if ϕ_j is in Background state. In this manner, ambiguities are solved, however the language introduces procedural steps not represented in the pictorial query and the interpretation of the query depends on both the pictorial representation and the drawing process.

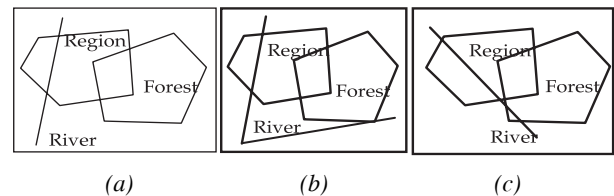


Fig. 1. Different pictorial queries.

The user checks the result translated into a textual form. He can thus avoid the generation of extraneous relationships. He can also impose a priori restrictive interpretations, before query processing and resolution.

Graphical representation of the query “Find all the regions which are *passed-through* by a river and *overlap* a forest” is shown in Figure 1-a, where the shape Forest must be in the background when the shape River is drawn (or, alternatively, the shape River must be in the background when the shape Forest is drawn).

In another approach, such as Spatial-Query-by-Sketch, the parser considers both the exact solution of the query and other approximate solutions obtained by removing or relaxing some constraints. For example, a topological relationship between two icons can be removed by also considering other nearest topological relationships as acceptable. This approach allows, for example, to consider as an acceptable result a pair of touching geographical objects, even if they are disjoint in the pictorial representation.

In Spatial-Query-by-Sketch, the query can be represented by one of the pictures of Figure 1 considered as near to the desired result. In fact, each query produces all results that can be obtained by relaxing a limited number of constraints (River disjoint Forest, in this example). Obviously, each query produces a set of results containing the one actually required, which can be selected by the user. Other types of ambiguities must be considered if a sketch represents the pictorial query. In fact, using this kind of approach, it is also necessary to recognize the different components that form the drawing. Such types of ambiguities are not discussed in this paper.

3. The Geographical Pictorial Query Language

The Geographical Pictorial Query Language (GeoPQL, [8]) is an evolution of the Pictorial Query Language (PQL, [9]) and resolves the previous limitations, improving and simplifying the human-computer interaction. It is possible to specify queries using *symbolic graphical objects* (*sgo*) that have the appearance of three classic types of shapes: point, polyline and polygon. The user can assign each *sgo* with a semantic linked to the different kinds of information (layer) in the geographical database. Specifically, constraints can be imposed on both the attributes of the geographical data and their topological position, and the query's target information (a specified layer or a set of layers) can be specified.

This makes it possible to formulate queries on the geographical database simply by drawing a spatial representation of symbolic graphical objects, without knowing its complex syntax or structure.

Paragraph 3.1 illustrates basic concepts of the pictorial query language, and paragraph 3.2 proposes the GeoPQL algebra.

3.1. Basic Concepts

An *sgo* is formally defined as a 4-tuple

$$\psi = \langle id, objclass, \Sigma, \Lambda \rangle$$

where:

- *id* is the identifier of the *sgo*;
- *objclass* is the set (possibly empty) of the classes iconized by ψ ;
- Σ (property set) represents the attributes to which the user can assign a set of values; this selection among the classes of objects or their instances, is iconized by ψ ;
- Λ is the ordered set of pairs (h, v) that defines the object's position with respect to a reference point in the working area.

From a geometric point of view, an *sgo* is defined by its border and its internal points.

Described below are semantics and properties of different operators, using the following symbolism:

$$\psi = sgo$$

$\partial\psi$ = geometric border of ψ , defined as:

- if ψ is a polygon, $\partial\psi$ is the set of its accumulation points (as defined in the set theory);
- if ψ is an open polyline $\partial\psi$ is formed by its extreme points;
- if ψ is a point $\partial\psi$ is the empty set.

$$\psi^o = \psi - \partial\psi = \text{geometrical interior of } \psi.$$

$$\text{Dim}(\psi) = \begin{cases} 0 & \text{if } \psi \text{ contains at least one point but} \\ & \text{not polylines and polygons} \\ 1 & \text{if } \psi \text{ contains at least one polyline} \\ & \text{but not polygons} \\ 2 & \text{if } \psi \text{ contains at least one polygon.} \end{cases}$$

The *sgo* and the null element together form the elements of the alphabet Λ . This alphabet, together with the operators defined above, form the reference model.

3.2. The GeoPQL Algebra

The GeoPQL algebra comprises 12 operators: Geo-union, Geo-difference, Geo-disjunction, Geo-touching, Geo-inclusion, Geo-crossing, Geo-pass-through, Geo-overlapping, Geo-equality, Geo-distance, Geo-any and Geo-alias. These operators, their semantics and the alphabet Λ , defined by the *sgo* <geo-point, geo-polyline, geo-polygon, geo-null> where “geo-null” defines the empty object, form the system's geographical model.

Below is a brief formal description of the above operators.

Geo-union definition (Uni): The Geo-union of two sgo ψ_i and ψ_j is a new sgo defined as the set of all points belonging to ψ_i and/or to ψ_j .

Formally: Let $\psi_i, \psi_j \in A$ be two sgo. Then:

$$\psi_i \text{ Uni } \psi_j \equiv \psi_h = \{x : x \in \psi_i \vee x \in \psi_j\}$$

and $\psi_h \in A$.

Geo-touching definition (Tch): two sgo ψ_i and ψ_j Geo-touch if the points common to the two ψ are all contained in the union of their boundaries. The result of this operation between ψ_i and ψ_j is a new ψ called ψ_h and defined by the set of points common to ψ_i and ψ_j .

Formally: Let $\psi_i, \psi_j \in A$ be two sgo. Then:

$$\psi_i \text{ Tch } \psi_j \equiv \psi_h = \{x : x \in \psi_i \wedge x \in \psi_j$$

$$\wedge x \in (\partial\psi_i \cup \partial\psi_j)\} \text{ and } \psi_h \in A.$$

Geo-inclusion definition (Inc): A symbolic feature ψ_i Geo-include another symbolic feature ψ_j (and we write $\psi_i \text{ Inc } \psi_j$) if all the points of ψ_j are also points of ψ_i . The result is a symbolic feature ψ_h that coincides with the second operand ψ_j .

Formally: Let $\psi_i, \psi_j \in A$ be two sgo, and let $(\psi_i \cap \psi_j = \psi_j) \wedge (\psi_i \cap \psi_j \neq \psi_i)$. Then:

$$\psi_i \text{ Inc } \psi_j \equiv \psi_h = \{x : x \in \psi_j \rightarrow x \in \psi_i\}$$

and $\psi_h \in A$.

Geo-disjunction definition (Dsj): Two sgo ψ_i and ψ_j are in Geo-disjunction (formally $\psi_i \text{ Dsj } \psi_j$) if the intersection of their borders AND the intersection of their internal points are both null.

Formally: Let $\psi_i, \psi_j \in A$ be two sgo and let

$$(\partial\psi_i \cap \partial\psi_j = \emptyset) \wedge (\psi_i^\circ \cap \psi_j^\circ = \emptyset)$$

$$\wedge (\partial\psi_i \cap \psi_j^\circ = \emptyset) \wedge (\psi_i^\circ \cap \partial\psi_j = \emptyset).$$

Then:

$$\psi_i \text{ Dsj } \psi_j \equiv \text{true}.$$

Geo-Pass-through definition (Pth): Let ψ_i be a polyline and let ψ_j be a polygon. Then, the operator Geo-pass-through is applicable if the polyline is partially inside the polygon (formally $\psi_i \text{ Pth } \psi_j$).

Formally: Let $\psi_i, \psi_j \in A$ be two sgo, where ψ_i is a polyline and ψ_j is a polygon, and let

$$(\psi_i^\circ \cap \psi_j \neq \emptyset) \wedge (\psi_i^\circ \cap \psi_j^\circ \neq \emptyset)$$

Then:

$$\psi_i \text{ Pth } \psi_j \equiv \psi_h = \{x : x \in \psi_i \cap \psi_j\}$$

and $\psi_h \in A$.

Geo-distance definition (Dst): Let $\psi_i, \psi_j \in A$ be two sgo of whatever type. Their Geo-distance is valuable and ≥ 0 if their intersection is null. The (minimum) distance $\text{Dst}(\phi = \min)$ between them is a numeric value representing this distance. This operator can be used to find all the sgo with distance θ (θ being one of the following symbols: $>, <, =, \leq, \geq, \neq$) from the reference sgo. The distance (δ_ϕ) value is given by: $\delta_\phi(\psi_i, \psi_j)_\theta = \psi_h$ where

- ψ_h indicates a bi-oriented segment representing the distance operator between ψ_i, ψ_j .
- ϕ is the qualifier which solves this ambiguity.
- θ is a selection expression that includes conventional operators ($>, <, =, \neq$, etc.) or methods that behave like operators.

Geo-difference definition (Dif): Let $\psi_i, \psi_j \in A$ be two sgo. The Geo-difference between two symbolic objects ψ_i and ψ_j is defined as a new sgo (ψ_h) which contains all the points which belong to ψ_i but not to ψ_j .

Formally: Let $\psi_i, \psi_j \in A$ be two sgo. Then, Dif:

$$\psi_i \text{ Dif } \psi_j := \psi_h = \{x : x \in \psi_i \wedge x \notin \psi_j\}$$

and $\psi_h \in A$.

Geo-crossing definition (Crs): Let $\psi_i, \psi_j \in A$ be two polylines, and let $\psi_i^\circ \cap \psi_j^\circ \neq \emptyset$. Then, Crs:

$$\psi_i \text{ Crs } \psi_j := \psi_h = \{x : x \in \psi_i^\circ \cap \psi_j^\circ\}$$

and $\psi_h \in A$.

Geo-overlapping Definition (Ovl): Let $\psi_i, \psi_j \in A$ be two sgo of the same type. Between them a non-null overlap exists if their intersection is still non-null and has the same dimension as the sgo.

Formally: Let $\psi_i, \psi_j \in A$ be two sgo of the same type (polyline or polygon) and let $\dim(\psi_h) = \dim(\psi_i) = \dim(\psi_j) \wedge (\psi_h \neq \psi_i) \wedge (\psi_h \neq \psi_j)$. Then, Ovl:

$$\psi_i \text{ Ovl } \psi_j := \psi_h = \{x : x \in \psi_i \wedge x \in \psi_j\}$$

and $\psi_h \in A$.

Geo-equality definition (Eql): Two symbolic graphical objects ψ_i and ψ_j are topologically equal if they are of the same type and have the same shape.

Formally: Let $\psi_i, \psi_j \in A$ be two sgo of the type and let $(\psi_i \cap \psi_j = \psi_j) \wedge (\psi_i \cap \psi_j = \psi_i)$. Then:

$$\psi_i \text{ Eql } \psi_j := \psi_h = \{x : x \in \psi_i \wedge x \in \psi_j\}$$

and $\psi_h \in A$.

Geo-any definition (Any): Let $\psi_i, \psi_j \in A$ be two sgo. Any admissible relationship between them is valid if the Geo-any operator is applied between them.

Formally: Let $\psi_i, \psi_j \in A$ be two sgo and let $\phi \in \mathcal{R}$ be an admissible operator between ψ_i and ψ_j and \mathcal{R} be the set of all admissible operators between ψ_i and ψ_j . Then:

$$(\psi_j) \leftarrow \text{Any} \rightarrow (\psi_i) \text{ if } \psi_i \phi \psi_j = \text{true}$$

$\forall \phi \in \mathcal{R}$.

The operator must be explicitly specified by the user, using the appropriate symbol.

Geo-alias Definition (Als): Let ψ_i be an sgo. ψ_j is an *alias* of ψ_i if it is the same sgo with a different shape.

The Geo-distance, the Geo-any and the Geo-alias operators must be explicitly specified by the user, using the appropriate symbol: $\leftarrow \rightarrow$ for the Distance operator and \leftarrow “relation name” \rightarrow for the Geo-any and the Geo-alias operators, (where “relation name” is *Any* or *Als*).

4. Resolution of Ambiguities

In a pictorial query representation, as shown in Figure 1, it is impossible not to explicitly represent at least one relationship between a pair of sgo, independently of whether a constraint must be represented in the query or not. For this reason, to obtain an unambiguous visual query

language, it must be possible to eliminate such constraints from the query’s pictorial representation. This is permitted by Geo-any. If such a relationship is defined between a pair of sgo, it means that no constraint exists between them. For example, consider once more the query: “Find all the regions which are *passed-through* by a river and *overlap* a forest”, where the user has no interest in the relationship between river and forest. Geo-any resolves some ambiguities relating to query formulation through different representations (see Figure 1), each having different constraints between forest and river.

Each query in Figure 1 can also be interpreted in different ways. The correct interpretation of Figure 1-a is “Find all the regions which are *passed-through* by a river and *overlap* a forest *and in which the river is disjointed from the forest*”. The correct interpretation of Figure 1-b is “Find all the regions which are *passed-through* by a river and *overlap* a forest *and in which the river touches the forest*”, while Figure 1-c is correctly interpreted as “Find all the regions which are *passed-through* by a river and *overlap* a forest *and in which the river passes-through the forest*”. By introducing the Geo-any operator, the query is correctly drawn as shown in Figure 2, and all relationships are correct, whether the river passes through the forest, is adjacent to the forest, or is disjointed from the forest. Note that the definition of the operator Geo-any also enables visually complicated queries to be avoided.

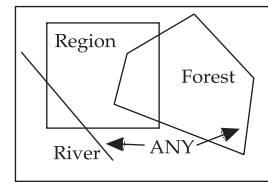


Fig. 2. The Geo-any operator.

Consider the formulation of queries with more than three objects (six, seven, etc.). The user considers what he basically requires, but does not consider all the *any* relationships existing between pairs in which he has no interest (i.e., *any* relationship is *good*). In fact, to represent all such relationships, the user would have to draw them all, resulting in an unmanageable situation. The Geo-any operator allows the query to be interpreted exactly in line with what the

user wishes to obtain as an answer. However, it is not always true that only one relationship exists between a pair of objects or that all of them are true. In many cases it must be possible to express one OR another given set of relationships. For this reason the Geo-alias operator is proposed. This allows a query to be implemented with the OR operator (in practice, the Geo-alias duplicates an sgo in order to draw a query in which it is used in the two alternative parts). Geo-alias also resolves some ambiguities which may arise in interpreting the query’s pictorial configuration. For example, Figure 1-b shows the pictorial representation of the query “Find the regions which are *passed-through* by a river OR which *include* a lake”. Without the G-alias operator, the query is ambiguous and may give rise to different interpretations.

Another problem in query interpretation derives from the possible need to consider the symbolic graphical objects obtained by applying operators to pairs of sgo and their mutual constraints, and then applying operators to the obtained sgo. The user does not want most of these constraints. For this reason, the only ones considered by the system are those derived from the drawing. Nevertheless, it is possible to transform a (virtual) sgo, obtained from the application of an operator to a pair of sgo, to an sgo directly drawn by the user and then to evaluate its constraints during query elaboration. In this manner, GeoPQL is able to manage and resolve the different situations shown in Figure 4. By calling the virtual symbolic object resulting from the operation “Region OVL Forest” X, it is possible to distinguish between the two situations: River Dsj X and River Pth X.

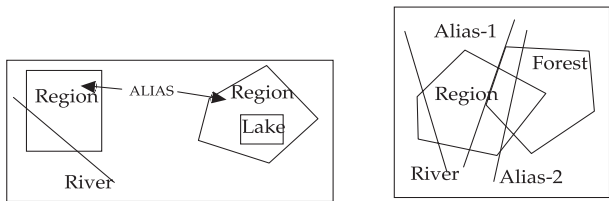


Fig. 3. The Geo-alias operator.

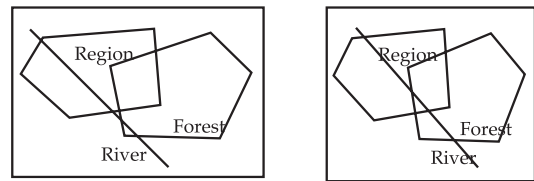


Fig. 4. Different constraints example.

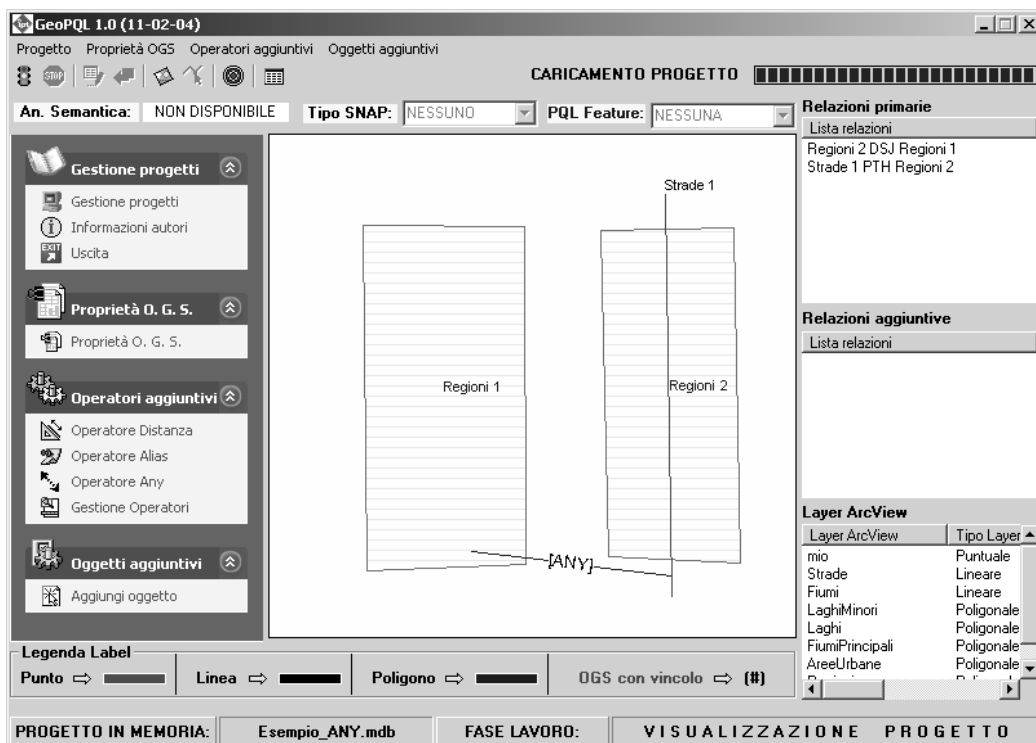


Fig. 5. A query example.

5. Pictorial Queries by GeoPQL

The implemented system interfaces with the ESRI GIS ArcView[®] and uses the same technology (ArcObjects). The layers stored in ArcMap, ArcView's database, are managed by GeoPQL using a symbolic association with the *sgo*, with which the user can define constraints on both the attributes of the geographical data and their spatial position.

The user formulates the query by drawing different symbolic graphical objects, assigning to each its semantic meaning and selecting the query target by clicking on the object (or part of it, or combinations of parts of objects). Before the target is defined by the user, the system writes all valid relationships between all possible pairs of previously drawn symbolic objects in a suitable window. When the target is defined, the system resolves any ambiguities which might arise in the interpretation. Figure 5 shows the pictorial representation of the query: <Find all "Region 2" which are *passed-through* by the free-way "Strade 1" AND are Disjoined from "Region 1">.

6. Conclusion

This paper presents the Pictorial Geographical Query Language GeoPQL, which enables easy query formulation. The main results obtained in GeoPQL are: a high expressive power (in fact, a user composes his query pictorially and thus expresses all topological relations among the geographic objects involved); the solution of ambiguities inherent to the spatial representation of a geographical query (in fact, it is possible to remove or modify some undesired constraints, using the operators introduced in GeoPQL) and an exact match between the query and obtained results (in fact, the results obtained are only those satisfying the constraints, with irrelevant constraints removed from the picture directly by the user).

References

- [1] MEYER B., "Beyond Icons: Towards New Metaphors for Visual Query Languages for Spatial Information Systems", *I^o Intern. Workshop on*

Interfaces in Database Systems, Springer-Verlag, pp. 113–135, 1993.

- [2] EGENHOFER M.J., "Query Processing in Spatial-Query-by-Sketch", *Journal of Visual Languages and Computing*, Vol. 8, N. 4, pp. 403–424, 1997.
- [3] MAINGUENAUD M., PORTIER M.A., "Definition of Cigales: a GIS Query Language", *Int. Conf. DEXA '90*, Springer Verlag Publ., pp. 275–280, 1990.
- [4] CALCINELLI D., MAINGUENAUD M., "Cigales, a visual language for geographic information system: the user interface", *Journal of Visual Languages and Computing*, Vol. 5, N. 2, pp. 113–132, 1994.
- [5] AUFAURES-PORTIER M.A., BONHOMME C., "A High Level Language for Spatial Data Management" *Third Int. Conf. on Visual Information Systems – VISUAL '99*, LNCS N. 1614, pp. 325–332, 1999.
- [6] AUFAURES-PORTIER M.A., "A High Level Interface Language for GIS", *Journal of Visual Languages and Computing*, Vol. 6, N. 2, pp. 167–182, 1995.
- [7] FAVETTA F., AUFAURE-PORTIER M.A. "About ambiguities in visual GIS query languages: a taxonomy and solutions", *VISUAL 2000*, LNCS N. 1929, Springer-Verlag Publ., pp. 154–165, 2000.
- [8] FERRI F., RAFANELLI M., "GeoPQL: a Geographical Pictorial Query Language", *Tech. Rep. IASI-CNR*, 2004 (in press).
- [9] FERRI F., MASSARI F., RAFANELLI M., "A Pictorial Query Language for Geographic Features in an Object-Oriented Environment", *Journal of Visual Languages and Computing*, Vol. 10, N. 6, pp. 641–671, Dec. 1999.

Received: June, 2004
Accepted: June, 2004

Contact address:

Fernando Ferri
IRPPS-CNR
via Nizza 128
00187 Roma
Italy
e-mail: f.ferri@irpps.cnr.it

Maurizio Rafanelli
IASI-CNR
viale Manzoni 30
00185 Roma
Italy
e-mail: rafanelli@iasi.cnr.it

DR. FERNANDO FERRI received the degrees in Electronic Engineering in 1990 and the PhD in Medical Informatics in 1994 from the University of Rome "La Sapienza". From 1993 to 2000 he was contract professor of "Sistemi di Elaborazione" at the University of Macerata and since 1996 has been a researcher at the Istituto di Ricerche sulla Popolazione e le Politiche Sociali of the Italian National Research Council. He is the author of more than 80 papers in international journals, books and conferences. His main methodological areas of interest are: data and knowledge bases, geographic information systems and human-computer interaction applied to different experimental areas, such as medical informatics and cultural heritage.

DR. MAURIZIO RAFANELLI, senior scientist at the Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti" (IASI) of the C.N.R. and in charge of the Section "Methods and Tools for Information and Knowledge Management". He was a part-time professor at the University of Rome – La Sapienza and at the University of Rome – Tor Vergata from 1980 to 1997. Author of over 100 international publications in journals, books, conferences, encyclopaedias, etc. His main methodological areas of interest are: data and knowledge bases, geographic information systems and human-computer interaction applied to different experimental areas, such as medical informatics, environment and pollution. He was editor of the book "Multidimensional Databases: Problems and Solutions" (Idea Group Publ., 2003) and General Chairman of the 4th and 10th International Conferences on Statistical and Scientific Database Management. In 1988 he was also at the Lawrence Berkeley Laboratory of the University of California as Fellowship Senior.
