ELECTRONIC **WORKSHOPS IN COMPUTING**

Series edited by Professor C.J. van Rijsbergen

Jessie Kennedy and Peter Barclay (Eds)

Interfaces to Databases (IDS-3)

Proceedings of the 3rd International Workshop on Interfaces to Databases, Napier University, Edinburgh, 8-10 July 1996

Paper:

Dynamic Maps: An Intuitive Interface for Naive Users of Spatial Database Systems

M. Dbouk, D. Kvedarauskas and P. Boursier

Published in collaboration with the
British Computer Society

Springer

Dynamic Maps : an Intuitive Interface for Naive Users of Spatial Database Systems

Mohamed Dbouk, Donatas Kvedarauskas, Patrice Boursier

University of Paris-Sud / Computer Science Research Laboratory
CNRS URA 410 & GDR 1041
LRI - Batiment 490 / 91405 ORSAY - FRANCE
Tel : +33 (1) 69.41.66.28 / Fax : +33 (1) 69.41.65.86
E-mail : {dbouk, dkvedar, patrice}@lri.fr

Abstract

Many approaches have been used for querying spatial databases, but they rarely distinguish between expert users and “naive” ones.

Two main approaches can be considered as interesting ways of querying and interacting with spatial and multimedia data, extended query languages on the one hand, and hypermedia techniques on the other hand. In this paper, we propose a framework for a navigational user interface model that will use both hypermedia and query language approaches.

Navigation methods will use two types of links, explicit hypermedia links on the one hand, implicit and dynamic ones on the other hand which are based on spatial relationships between entities.

We also propose different kinds of navigation strategies that will be usable and hopefully accepted by general public-like users.

1 Introduction

Data which are used in *geographical information systems* (GIS) are rich in nature, due to the spatial part or character of these objects. Now, if we consider the whole set of spatially-oriented GIS applications, we must differentiate between two kinds of users, expert users on the one hand and non-experienced or naive ones on the other hand.

Extended relational and object-oriented query languages are considered as powerful tools for manipulating and querying complex data sets in general, and they can be used as such by the first kind of users. Unfortunately, it is not the case for the second kind of users. The proposition of Peloux and Rigaux [23] can be considered as an interesting approach using OO-DBMS, but the user interface is still complex for a naive user.

Techniques based on *visual languages*, like Cigales [6], Sketch or Iconic [19, 20], provide a set of icons with which the user composes queries. The main drawbacks come from the complexity of the user interface, the need for standards and the expression of negation clauses.

Hypermedia techniques can simplify the use of complex and visual data [2, 18]. Navigation represents the basic and essential notion that characterizes this approach. Now, there are some drawbacks that come from the limited access to information, since the use of hypermedia-based information systems is conditioned by the existence of predefined links between objects. This method can yet be considered as the best way to help non-experienced users. In [25], an hypermedia approach is proposed for generating meta-information structures using recursive methods. It will be oriented towards professional designers of user interfaces.

Another approach by Chen and Mathé [9] is also interesting since it uses several cognitive concepts. However, the use of the context should be more developed in order to take into account the intellectual capability of users, a naive user being able to become after some time an experienced one.

In summary, the complexity of methods and techniques for querying a multimedia or hypermedia GIS depends on the goals and awaited results, and it is obviously difficult to satisfy all kinds of users. If the user is non experienced and he wants to take advantage from the richness of spatial data, then what is the best and possible solution that we can provide to him ? The query language is a complex solution, hypermedia techniques are not sufficient, and visual or iconic languages must be improved [4, 5].

The conclusion comes that the use of spatial information by non experienced users requires a multi-disciplinary approach, combining object-orientation, hypermedia and visual query languages. An interesting non spatial approach is used in [14]. Specific features have also been proposed in order to access spatial information with user controlled interaction [3, 7]. The propositions from de Hoop and van Oosterom [16, 22] are also very interesting, since they are based on the use of scale properties in GIS. The user interface model contains complex features (set queries for example), but the use of index techniques can be considered as a positive point. Carlotto [8] proposes another approach, more complex for naive users, in which a Macintosh style is used to develop HyperMap models.

Following these considerations, we propose a user interface model which combines the capabilities of the various approaches described above, and provides a simple environment for presenting and interacting with geographical entities through what we have called *dynamic maps*. This approach is based on the HyperGeo data model [11, 12, 13] .

In the following sections, we introduce the architecture of the HyperGeo prototype and the underlying logical data model of HyperGeo. We then give details about the components of the user interface, the architecture and the operational ways. We show the advantage of using at the same time hypermedia links and spatial relationships. We also show how we can navigate using them, and finally how we can backtrack using the historic manager.

2 Introduction to HyperGeo

2.1 Functional Architecture of HyperGeo

Most of existing GIS tools, either commercial products or research prototypes, are based on the use of a relational DBMS for handling geographical data. Some of them, for example ESRI's ARC/INFO software, rely on a *coupling* approach as a relational DBMS is used for handling the descriptive part of geographical data, and a specific component is used for the spatial part. Others are based on a *layered* approach as a geographic layer is specifically designed over a relational DBMS. The additional layer is sometimes based on an obect-oriented language, as is for example the case with the SMALLWORLD GIS software.

The HyperGeo system takes another approach since it is based on the use of an object-oriented DBMS and its query language with an additional hypermedia layer. More precisely, it has been designed around three layers (cf. Figure 1) : a navigational user interface layer, which is the subject of this paper, a query language layer, and a data storage layer (database level).

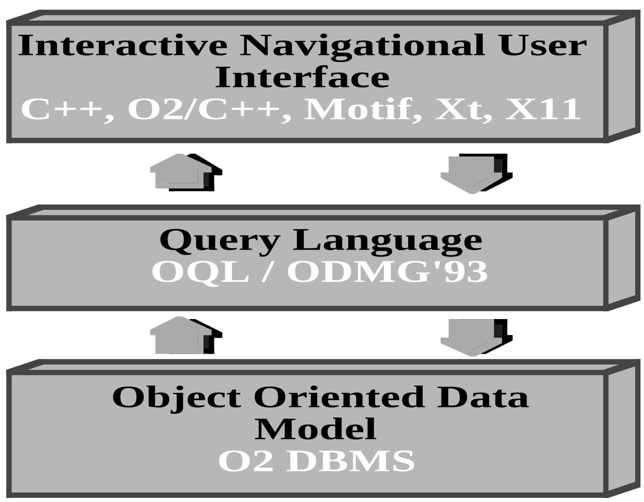


Figure 1 : Architecture of HyperGeo

The HyperGeo data model was implemented on top of the O2 object-oriented database management system [1], with the OQL query language allowing to access data. For this task, we use the appropriate *o2query* function [21], and users can thus query data transparently. But the query language layer in the HyperGeo architecture also makes possible GIS interoperability with the querying of geographical data sets distributed among heterogeneous geographical information systems [17].

2.2 Introduction to the HyperGeo Data Model

From a data modelling viewpoint, the observed universe is considered as a set of heterogeneous *entities* like regions, cities, streets, buildings, lakes or roads. All these entities have a spatial component, and they are connected by different kinds of *links*. Consequently, these two basic concepts, entities and links, are used for describing the real world.

Entities have a non-spatial (or descriptive) part, and a spatial part that can be represented by one or several digital *spatial representations* coming from the digitization of maps, possibly at different scales. These spatial representations may be general, or specific of some kind of some application (e.g. topologic, network, ...).

In addition, since we address general public applications, some *multimedia representations* may be associated to geographic entities. For example, photographies of monuments, touristic videos or satellite images may be associated to the city of Paris.

In the declarations that follow, object identifiers are explicitly declared within classes of objects. When using the O2 DBMS, they become transparent since they are declared and handled automatically by the system.

```
hg_Entity : class    // explicit format of generic class of HyperGeo entities
     ID : Object_Identifier,
     SR : set (Spatial_Representation : hg_Representation),
     VR : set (Multimedia_Representation :  hg_Representation),
     CL : set (Composition_Link : hg_Link),
     HL : set (Hypermedia_Link : hg_Link),
     SR : set (Spatial_Relationship : hg_Link).
     ...
Spatial_Representation : class
     ID : Object_Identifier,
     Source : Map_Source,
     Scale : Map_Scale,
     Date : Map_Date,
     Type : SR_Type
     ...
Multimedia_Representation : class
     ID : Object_Identifier,
     Type : Multimedia_Type,
     ...
Image : class inherits Multimedia_Representation
     <specialization>
Video : class inherits Multimedia_Representation
     <specialization>
```

Specific classes of semantic entities, for example hg_Town or hg_Road, will inherit from hg_Entity. They will be specialized in consequence.

```
hg_Town : class inherits hg_Entity
     DA : tuple (Descriptive_Attributes),
     <specialization>
hg_Road : class inherits hg_Entity
     DA : tuple (Descriptive_Attributes),
     <specialization>
```

Links constitute the second basic class of components in HyperGeo. We will distinguish *explicit links*, that consist of both hypermedia links and composition links and are stored into the database, and *implicit links* which are computed from the geographical coordinates of objects when required.

Composition links (CL) are used to represent such composite objects as cities divided into quarters or countries divided into regions. *Hypermedia links* (HL) reflect the hypermedia orientation of the system. They allow to associate multimedia representations to geographical entities both at the data model and at the user interface levels.

Spatial relationships reflect the spatial component of geographical data. All geographic entities are linked by such relations that concern : (i) distance measurement between entities, (ii) topological relationships (e.g. inclusion, intersection or adjacency), and (iii) relative positioning between entities (north, east, ...). Spatial relationships are not explicitly but only virtually stored since they are computed from the geographical coordinates that constitute the spatial part of entities.

Links will be also conditioned, i.e. they will have a context inductor that depends on the representation context. For example, some links associated to the “Paris” entity may be visible at the Paris city level and at the Ile-de-France region level, but neither at the France country level nor at the European level.

```
hg_Link : class                // generic class of HyperGeo links
     Destination : hg_Entity,
     Type : Link_Type,
     Context : Link_Context,
...
Composition_Link : class inherits hg_Link
     <specialization>
Hypermedia_Link : class inherits hg_Link
     <specialization>
Spatial_Relationship : class inherits hg_Link
     <specialization>
```

Figures 2 and 3 illustrate and summarize the main concepts of the HyperGeo [13] data model that were presented above.

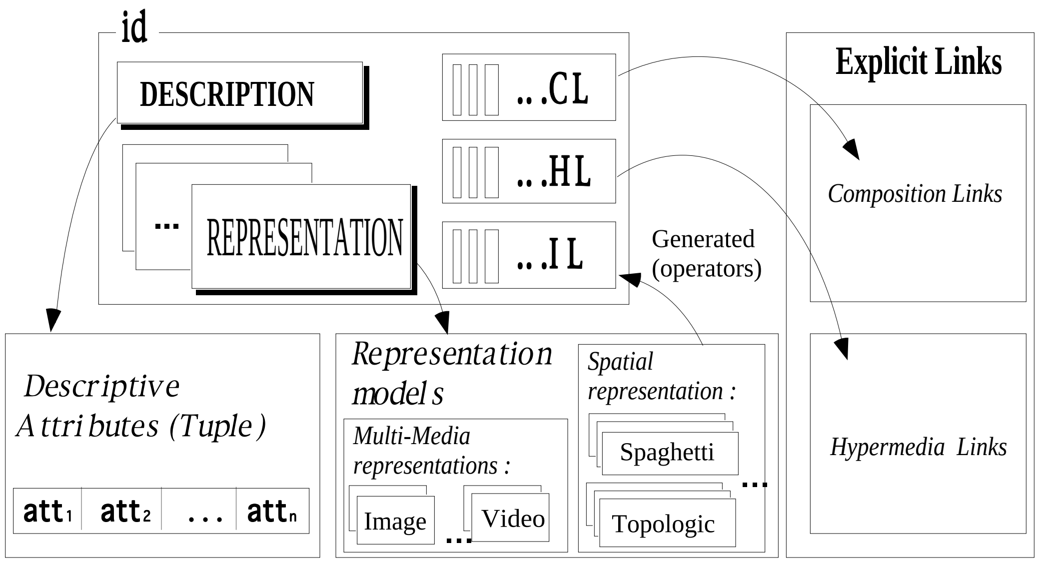


Figure 2 : Conceptual Schema of a Generic HyperGeo Entity

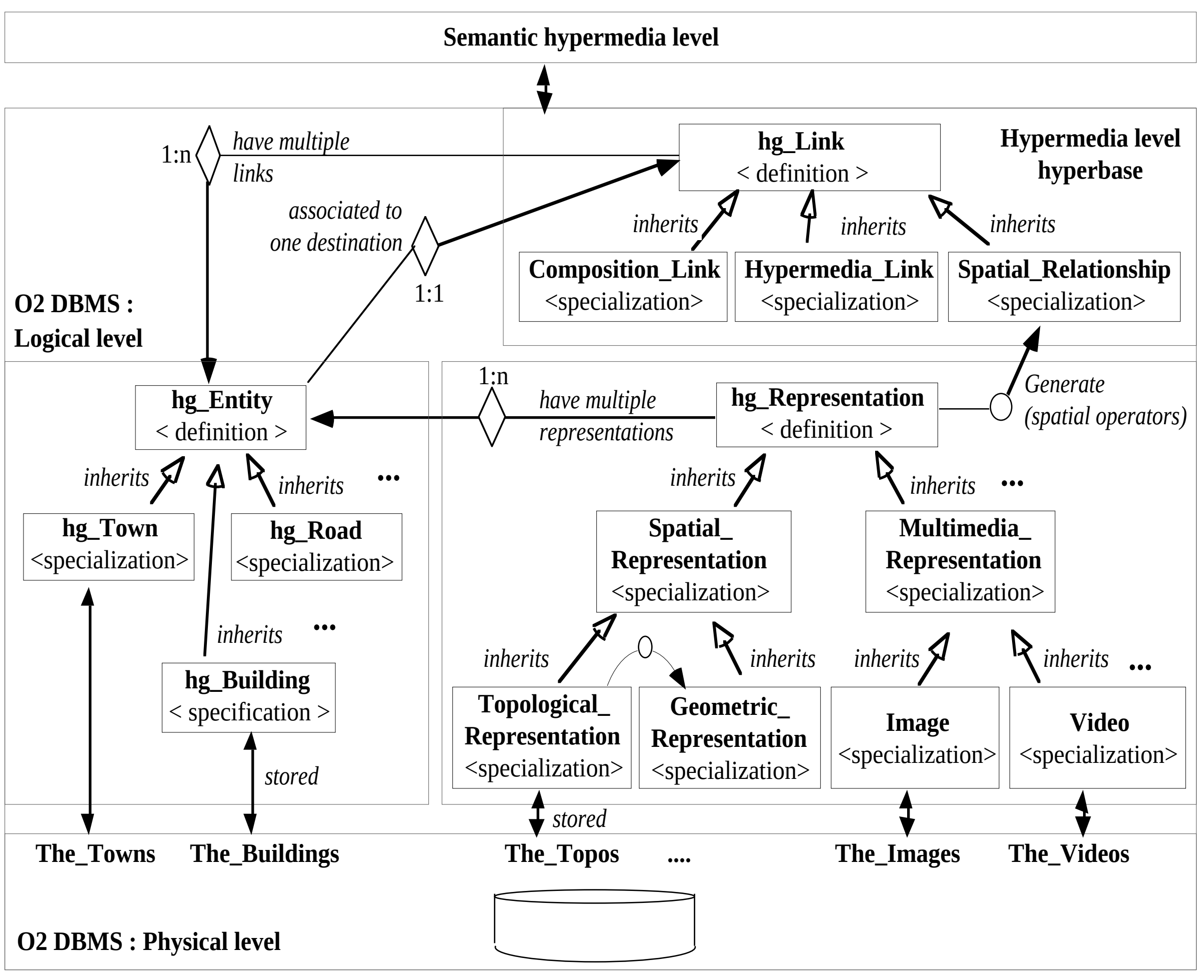


Figure 3 : HyperGeo Data Model

2.3 Dynamic Maps

In addition to these concepts that describe geographical entities and links, we must also introduce the new and basic concept of *dynamic map*. A dynamic map corresponds to a geographical data set that is extracted from one or several spatial databases with a spatial query and presented to the user. This data set is characterized by a limited geographic area or territory, a given scale and a set of selected *themes* (or geographic data types), but it is also characterized by the fact (inherent in the HyperGeo data model) that selected entities have associated links that make the map navigable through explicit links and spatial relationships (cf. following sections).

Navigating within this data set, or changing the presentation (e.g. colors or displayed themes) will not affect the dynamic map itself, but only the presentation that is displayed and shown to the user. A new dynamic map will be built only when a new geographic area or new themes are required.

The map can be displayed in a 2D standard presentation, but it may also be displayed using 3D models when digital elevation model (DEM) data are available and associated to the map. The scale and projection may also be changed.

```
hg_Dynamic_Map : class
     Area : Map_Area,Scale : Map_Scale,
     Projection : Map_Projection,
     Key : Map_Key,
     Presentation : Map_Presentation,
     Entities : set (hg_Entity).
```

A more complete presentation of this concept will be given in section 4.1 for what concerns more precisely the user interface and navigation aspects.

3 The HyperGeo User Interface Model

The approaches that have been mentioned in the introduction give answers to some problems, but they also leave others unsolved. Query languages favor experimented users, but they imply learning and they require a good mastery of the syntax. With this technique, we can also use spatial relationships between entities. On the other hand, hypermedia methods favor naive users, and they represent the basic method for really interacting with entities, but they lack expressive power.

The HyperGeo user interface combines these approaches and allows naive users to use in the same environment implicit links (via spatial relationships) and explicit ones (composition and hypermedia links) to navigate and use semantic associations that may exist between entities. They can thus act as if they were querying the database, but with a very limited knowledge of its structure, and without using a real query language.

The user interface also provides simple and complete display utilities. The main services given by this approach are : (i) to give at any time the look of objects using *map metaphors*, and (ii) to show results of queries in real time.

3.1 Functional Architecture

The conceptual model of this interface inherits from the PAC model [3,10]. Consequently, the architecture of the user interface consists of two main components, the *Map Manager* and the *Interaction Manager.*

The Map Manager takes care of maps to be displayed, allowing to show selected entities, but also to add or delete themes. This component uses a map metaphor that allows to define dynamic maps (cf. section 2.3).

The Interaction Manager controls any user interaction like selecting objects, windowing, or composing simple queries. It also controls the import and export of data objects, as well as user transactions between the user interface level and the database one. It contains three sub-managers, namely the *Dialog Manager*, the *Transaction Manager* (or query manager) and the *History Manager*. In fact, we define two dialog modes : the interaction or navigation mode, and the query (or transactional) mode. Transactions (accesses to the database) will be handled by the Transaction Manager.

The Dialog Manager will supervise several dialog windows : the navigation window, the themes and operators window, and the representations window.

The History Manager consists of configurable components that will be used to store maps automatically, and to backtrack through a web of previously visited maps.

3.2 Ergonomic Architecture

Choosing a simple look is a performance criterion when designing a user interface. The HyperGeo system offers an easy-to-use user interface and adopts a one display presentation model (2D) instead of a nesting one (2D1/2). This display behaves according to HyperGeo interface managers. Figure 4 shows the visual components of this interface.

Components communicate through corresponding managers and share some sets of entities in order to ensure the consistency of the interface environment.

For example, the Map Manager an the Dialog Manager share active maps, obects of the hg_Dynamic_Map class. The Transaction Manager shares objects with the Map Manager when we compose queries concerning the database.

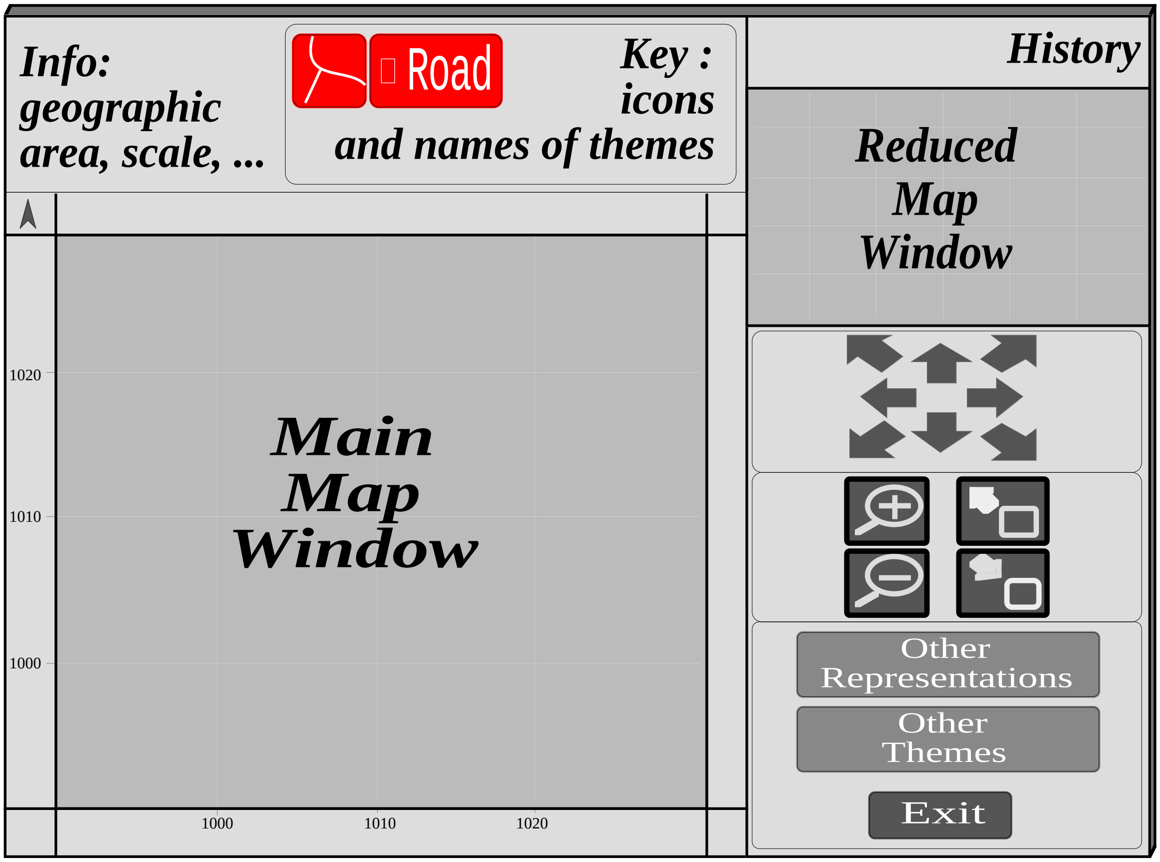


Figure 4 : Ergonomic View of the HyperGeo User Interface

3.3 Functionalities

The HyperGeo user interface has been designed with one main objective : make possible the access and use of spatial and multimedia complex data by naive users, using a reduced set of simple interaction methods.

For that purpose, it offers the following functions :

- show spatial and multimedia complex entities (e.g. vector maps, images, sounds, 3D representations) using graphical projection operators,
- query entities by interaction using hypermedia techniques (i.e. mouse pointing, windowing), and simple (atomic) queries according to the representation models (e.g. topologic) which are used,
- navigate through selected entities, using explicit links (multi-level and multi-scale, hypermedia anchors) and implicit ones (windowing and changing level and scale, spatial relationships),
- enable backtracking capabilities : when the user navigates through semantic entities, the History Manager transparently stores all the maps that have been previously visited,
- handle a graphical context for any HyperGeo component, using predefined meta-information,
- handle the multi-representation of entities.

4 Navigation within Dynamic Maps

In HyperGeo, the navigation mode is based on interaction capabilities with semantic entities. Then, the look of the user interface, and especially the choice of metaphors, will be in conjunction with the aim of reducing the duration of the perception of tasks by users. In other words, it must be adapted to the skill level of users.

4.1 The Choice of Metaphors

With the HyperGeo user interface, different kinds of metaphors are used to present data corresponding to various tasks [11, 12, 20]. For example, we use the reactive dynamic map metaphor to visualize some selected set of entities in the main map window.

A dynamic map can be defined as follows :

- a set of heterogeneous spatial and multimedia reactive entities,
- an evolutive content using hypermedia navigation and database queries,
- multiple representation models that can participate to the composition of a map,
- a set of visual information associated to a map.

Associative metaphors (e.g. buttons or icons for hypermedia anchors) are used, in order to enable interaction with spatial operators like pointing, distance, or intersection. For example, we use a menu of buttons or icons (on/off) metaphor to load the corresponding dialog windows (e.g. operators) and the key window of the active map. Organizational metaphors are also used to present explicit links based on the geometry of entities.

Metaphors must reflect the traditional and popular concepts well known by aimed users, such as maps, pictograms or images (cf. Figure 5). Any break with the daily life of users must be avoided, and this represents one major criterion among the conditions for the acceptance or the reject of the system.

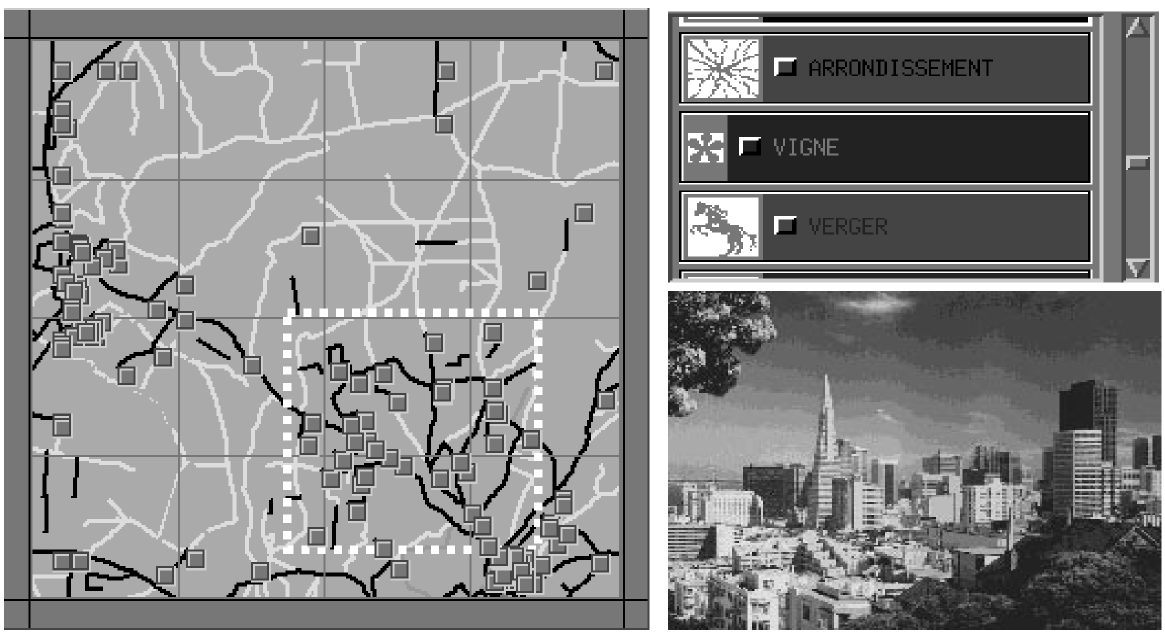


Figure 5 : Metaphor Styles

4.2 User Interaction with Maps

The HyperGeo user interface presents information concerning a set of selected entities using the dynamic map metaphor, with operators and dynamic key information. It distinguishes two modes for querying dynamic maps : dialog (interaction) mode, and querying (transactional) mode.

In dialog mode, users have two ways for interacting with displayed information : selecting entities directly, using the mouse in the main or reduced window, or using navigation operators. Among possible interactions, we propose point selection and rectangle selection.

4.2.1 Selection in the Main Map Window

Users can select one or multiple entities (anchor, button, or vector-based geographical entity) by clicking in the main map window. Then, a navigation control window appears in the place of the dialog one. The user can see the description of the selected entity, zoom on it, use explicit links (hypermedia or composition links) or call for other map representations when it is possible (e.g. a 3D representation when DEM data are available). He can also use implicit links (spatial relationships) by calling the corresponding operator (e.g. distance or intersection between a town and a highway). The selected entity (or entities) will be redisplayed in the main map window with another graphical context, and the key window will be changed.

Consequently, such a *point selection* operation in the main map window generates an update of all the interface components. In any case, if the user has clicked twice on a button, the destination entity will appear in the main map window.

This method of navigation, using hypermedia techniques and database queries, is adapted to help naive users by apprenticeship. It is controlled by the modules of the Interaction Manager.

It is also possible to select a set of entities in the main map window by specifying a geographical selection area (rectangle) with the mouse (*rectangle selection*). To calculate the result, the Dialog Manager activates a spatial *o2query* function to the O2 database.

4.2.2 Selection in the Reduced Map Window

The reduced map window represents a global view (at a smaller scale) of the geographic area which is presented in the main map window. Users can also interact within the reduced map window by clicking on it (point selection) or defining a selection area (rectangle selection).

A point selection operation in this window allows the replacement of the main map by the region corresponding to the selected object. A rectangle selection operation allows the same result, but in that case the selected rectangle is used to generate the corresponding region.

4.2.3 Dialog Window

A set of navigation operators is proposed in this window. We have distinguished the following rules to interpret user motions illustrated by Figure 6:

- horizontal motion, to east, west, north, north-west, etc., (connectivity in 4 or 8 directions),
- vertical motion, from the top to the bottom, or in the opposite direction,
- arbitrary motion, by jump in any direction (destination can be fixed).

Linear adjacent navigation represents horizontal motion, the user being able to move at one level without changing environment features (cf. Figure 7). For that he will intuitively use the eight direction buttons. A configurable algorithm will be called by the Dialog Manager to calculate the new map (vectorial entities). This algorithm can use eight buffers, one for every direction (buffers will be generated transparently in real time). Point selection in the reduced map window allows *linear non adjacent* navigation (jump).

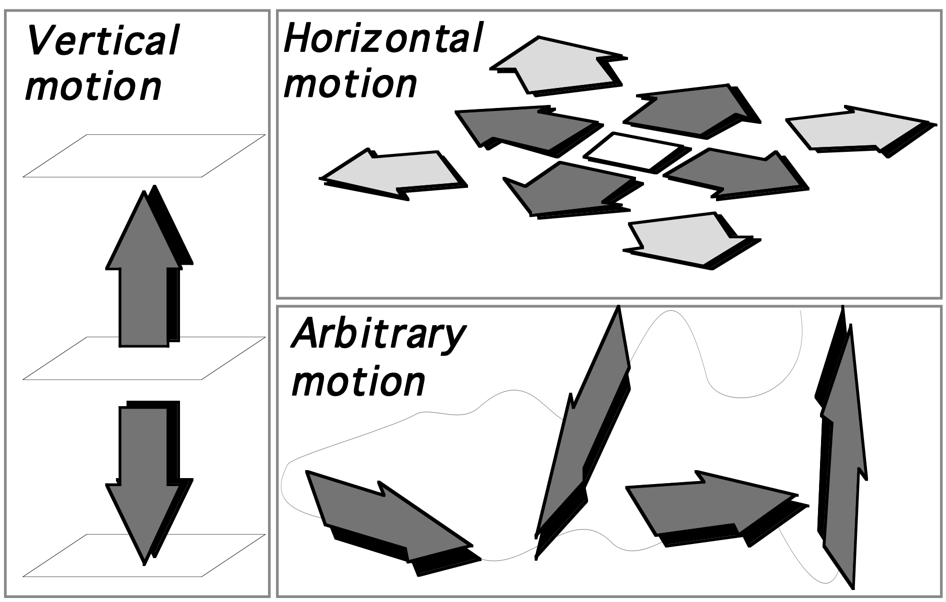


Figure 6 : Possible Motions

The second category of motions can be modelled using zoom operations (cf. Figure 8). For that, we propose two different functions. An “intelligent” (or logical) zoom will change the level (scale), and new entities will appear after querying the database and refreshing the display. A “geometric” (or physical) zoom is also offered, which only redisplays at a different scale without looking for more information. The Map Manager and the Dialog Manager use the multiple selection method to calculate results.

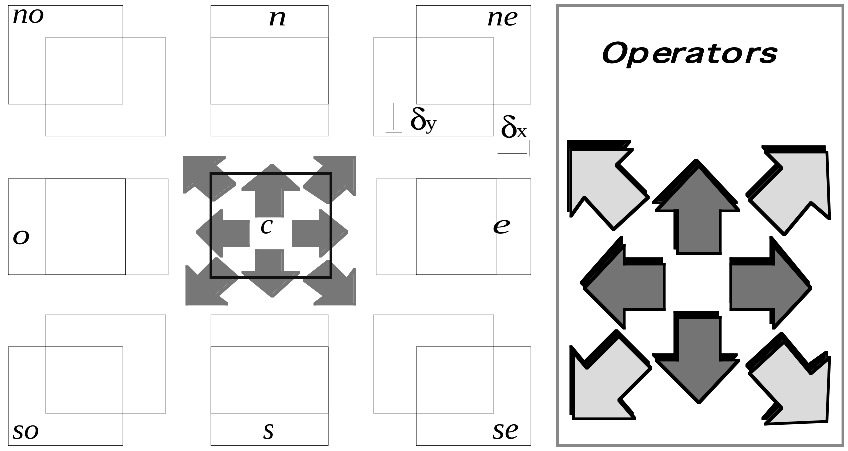


Figure 7 : Horizontal Navigation Features

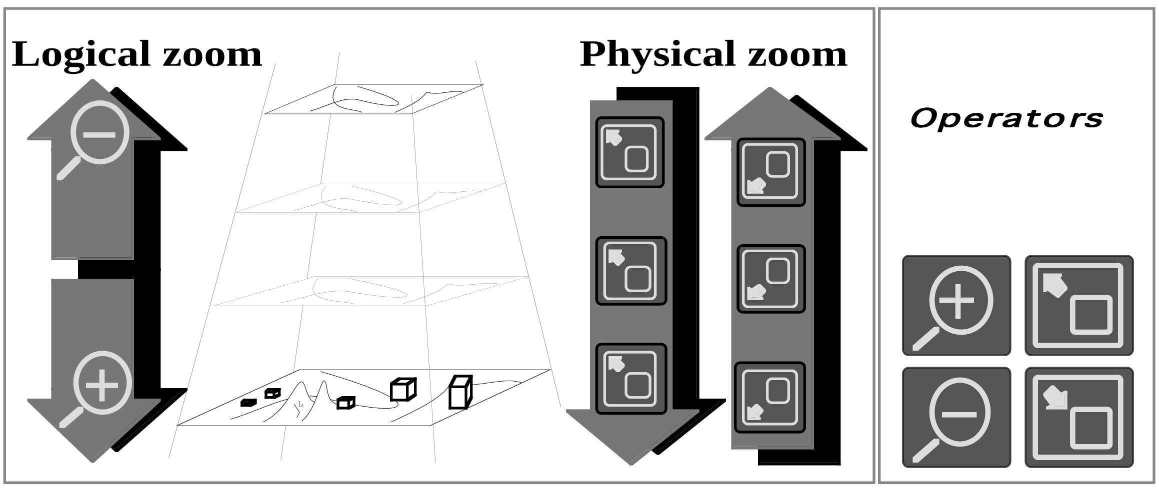


Figure 8 : Vertical Navigation Features

Finally, arbitrary motion may concern linear adjacent operation (east, west, north, south), or navigation using explicit links. The destination of an explicit link can be at any level and at any place. Implicit links can select a set of non adjacent entities stored in a buffer map and rewritten in the main map changing the graphical context. A pointing vertical zoom operation can also be used. Selected entities can be called from the main map or from the database using transaction operation by the Dialog Manager.

4.3 Sample User Interaction

The HyperGeo user interface will simplify the use of complex methods for querying spatial data, as is shown in the example given in Figure 9 (only part of the display is shown). In this example, a point selection operation in the main map window first allows to select a set of entities located at the same place in the map and that belong to different themes, or thematic layers (e.g. forests and roads). They are listed in the Dialog Window, and the user can choose one entity from this set (cf. Figure 9a). When he clicks on one entity, it is redisplayed in real time in the main map window with a new visual context. At the same time, dialog properties attached to this entity also appear in the dialog window, i.e. hypermedia links, composition links, other representations if they are available, as well as spatial relationships operators like distance, intersection or adjacency (cf. Figure 9b).

The user can then select one operator that will generate and evaluate a set of temporary spatial links between the selected entity and others which are stored into the database. In the example of Figure 9, the distance operator has been chosen, and consequently the user will have to specify, in the dialog window, the distance limits and the set of themes which are concerned (cf. Figure 9c).

During all this process of user interaction, the dialog manager composes transparently the corresponding queries using *o2query* functions :

a) Point selection -> Q1 :

```
select    entity
from      entity in current_map->Entities
where     entity->SR[current]->pointing(selected_entity->SR[current], Mouse)=true
```

b) Selection of the entity “selected_entity” from the result of Q1 displayed in the Dialog Window,

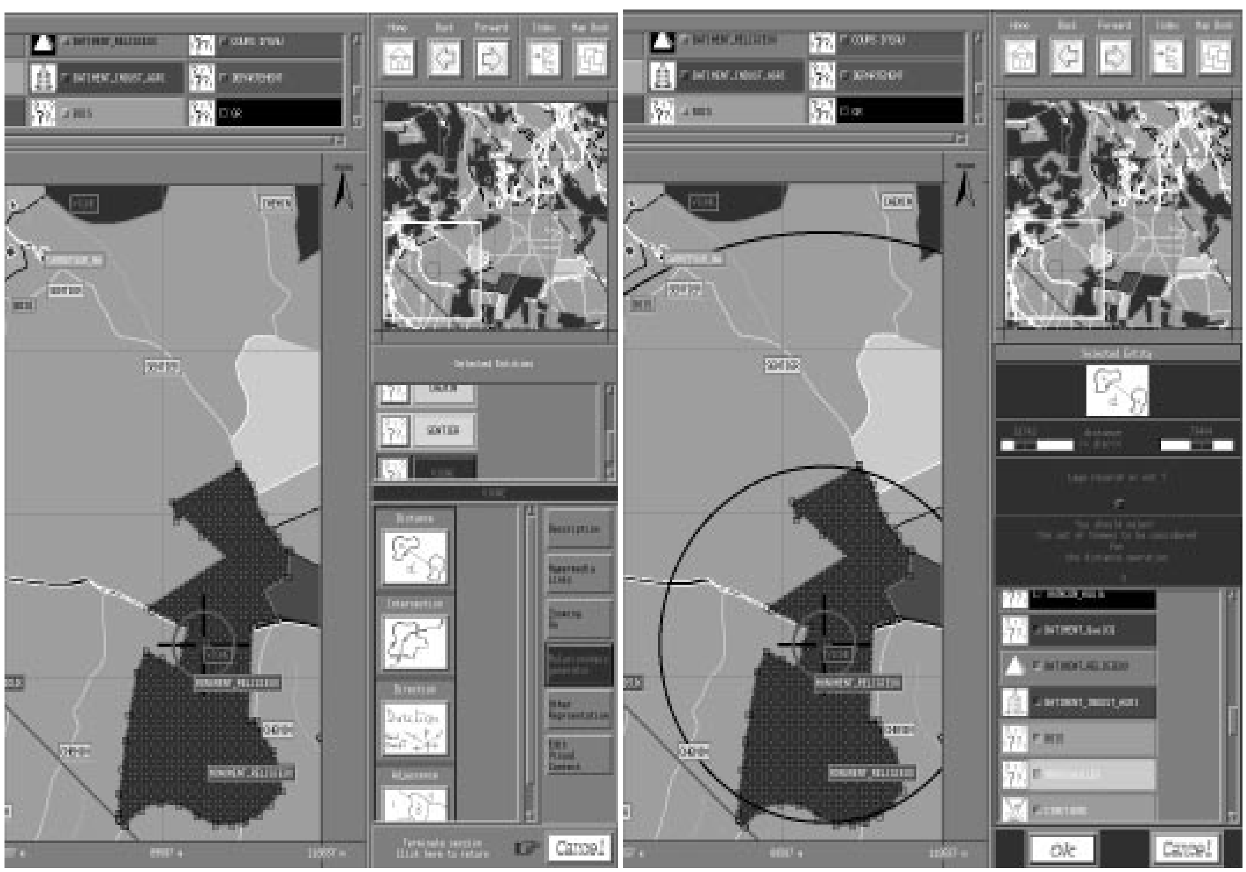
c) Selection of the operator “distance” from the set of operators associated to selected_entity,

d) Distance query processing -> Q2 :

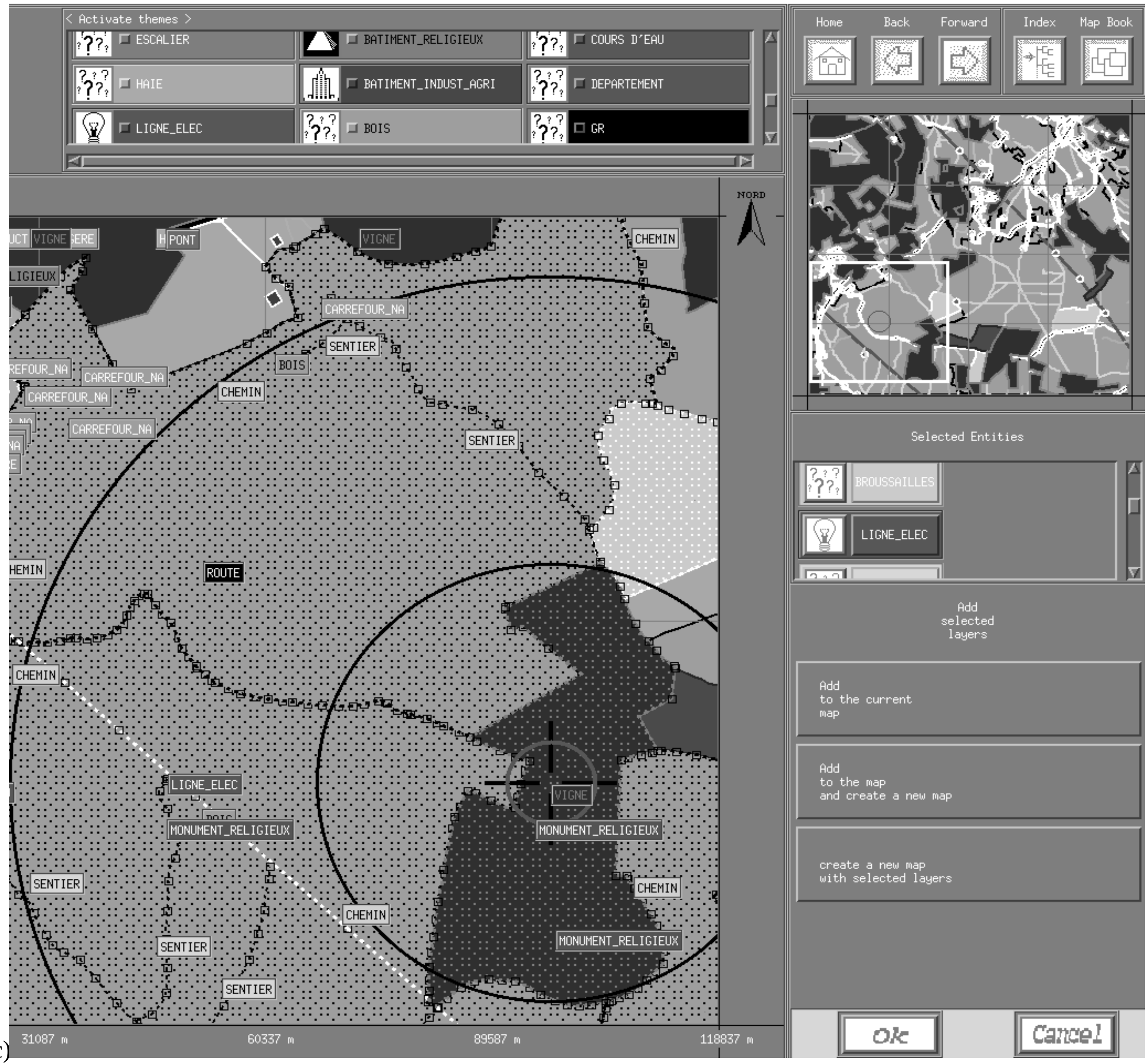
```
select    entity
from      entity in considered_set_of_entities
where     entity->SR[current]->distance(selected_entity->SR[current], distance_low,
          distance_high)=true
```

The entities resulting from Q2 represent the set of entities associated to selected_entity by query Q2. This new link will be typed, the type being the distance operator. These entities will be redisplayed in the main map window (cf. Figure 9c), and the user can start from this point for a new interaction or for preparing a new query to the database.

In summary, the user can compose single or multiple queries in an iterative (step by step) and interactive way, using successive selection and filtering operations.



(a) (b)



c)

Figure 9 : Sample User Interaction

5 Conclusion

The HyperGeo user interface combines both hypermedia techniques and database query language concepts. It is navigational, oriented towards general public-like naive users. It also gives a simple presentation display with all useful information (contextual key). Then, it offers two consultation modes : dialog (or interaction) mode and query (or transactional) one.

Users can navigate using the mouse by pointing/selection operations. They can use different navigation strategies like, adjacent linear, vertical (zoom operations) and other (arbitrary) possibilities. HyperGeo distinguishes explicit links (hypermedia and composition) and implicit ones (spatial relationships). A user can discover information about maps step-by-step by apprenticeship. On the other hand, a presentation controller improves the richness of maps.

Finally, this approach represents a user interface model making possible the use of complex spatial and multimedia data by non-experienced users communities. It has been developed on top of X-Window/Motif, using the O2 object-oriented database management system, and validated using a set of data coming from the french geographical institute (IGN).

Further developements concern the use of the user context for improving interactive navigation and querying. We also work in the field of machine-learning for the system to adapt to the evolution of user's knowledge and skill.

References

1. Bancilhon F, Delobel C, Kanellakis P. (eds). Building an objet-oriented database system : the story of O2. Morgan Kaufman, 1992.

2. Berk E., Devlin J., "HyperText / HyperMedia Handbook", Academic Press, 1990.

3. Booth P. An Introduction to Human-Computer Interaction. Lawrence Erlbaum Ass., UK, 1991.

4. Boursier P. Hypermedia-based geographical information handling. Proc. Second European Conf. on Geographical Information Systems (EGIS '90), Brussels, 1990.

5. Boursier P, Mainguenaud M. Spatial query languages : extended SQL vs. Visual Query Languages vs. Hypermaps. Proc. Fifth Int. Conf. on Spatial Data Handling, Charleston, VA, USA, 1992.

6. Calcinelli D, Mainguenaud M. CIGALES, a Visual Query Language for Geographical Information System : The User Interface. Journal of Visual Languages and Computing, Vol. 5, Academic Press, 1994.

7. Card SK, Moran TP, Newell A. The Psychology of Human-Computer Interaction. Lawrence Erlbaum Ass., 1983.

8. Carlotto MJ. Text attributes and processing techniques in geographical information systems. Int. Journal GIS, Vol. 9, No. 6, 1995.

9. Chen JR, Mathé N. Learning Subjective Relevance to Facilitate Information Access. Proc. Fouth International Conference on Information and Knowledge Management (ACM-CIKM' 95), Baltimore, USA, 1995.

10. Coutaz J. Interfaces Homme-ordinateur : Conception et réalisation. Dunod Infomatique, 1990.

11. Dbouk M, Boursier P. HyperGeo : a Hypermedia-based Geographical Information System Design Framework. Proc. Joint European Conference on Geographical Information Systems (JEC-GI '95), The Hague, The Netherlands, 1995.

12. Dbouk M., Boursier P., "Dynamic Maps for Navigating through Geographical Data Sets", ACM Workshop on New Paradigms for Information Visualization and Manipulation, Baltimore, USA, December 1995.

13. Dbouk M, Kvedarauskas D, Boursier P. A Multi-Representation and Multi-Scale Data Model for Hypermedia GIS Applications. Proc. Int. Symposium on GIS/Remote Sensing Research, Development, and Applications (GeoInformatics '96), Florida, USA, 1996.

14. Duce DA, Gomes MR, Hopgood FRA, Lee JR. (eds). User Interface Management and Design. Proc. Workshop on User Interface Management Systems and Environments, Springer-Verlag, 1990.

15. Heller D. Motif Programming Manual, Vol. 6, Rel. 1.1. O'Reilly & Associates, 1991.

16. de Hoop S, van Oosterom P, Molenaar M. Topological Querying of Multiple Map Layers. Proc. Int. Conf. on Spatial Information Theory (COSIT '93, Lecture Notes in Computer Science, No. 716), 1993.

17. Jen TY, Dbouk M. The use of Spatial Relationships in a Hypermedia-based Geographical Information System. Proc. Int. Symposium on GIS/Remote Sensing : Research, Development, and Applications (GeoInformatics '96), Florida, USA, 1996.

18 Laurini R., Thompson D., "Fundamentals of spatial information systems", McGraw-Hill, 1992.

19. Lee YC, Chin FL. An iconic language for topological relationships in GIS. Int. Journal Geographical Information Systems, Vol. 9, No. 1, 1995, pp. 25-46.

20. Meyer B. Beyond Icons : Towards New Metaphors for Visual Query Languages for Spatial Information Systems. Proc. First Int. Workshop on Interfaces to Databases Systems, Springer-Verlag, Glasgow, UK, 1992.

21. O2 Interface to C++, Rel. 4.5, O2 Technology, 1995.

22. van Oosterom P, Schenkelaars V. The development of an interactive multi-scale GIS. Int. Journal GIS, Vol. 9, N°5, 1995.

23. Peloux JP, Rigaux P. A Loosely Coupled Interface to an Object-Oriented Geographic Database. Proc. Int. Conf. on Spatial Information Theory (COSIT '95, Lecture Notes in Computer Science, No. 988), 1995.

24. Väänänen KA. Metaphor-based Authoring Tool for Multimedia Environments. Proc. Human Computer Interaction Vienna Conf. (VCHCI '93, Lecture Notes in Computer Science), 1993.

25. Valkovsky B, et al. The Processing of Recursive Generalization in the Construction of Hypermedia Macrostructure. Proc. Int. Conf. on Human-Computer Interaction (EWHCI ’95, Lecture Notes in Computer Science, No. 1015), Russia, 1995.