A Direct Manipulation User Interface for Querying Geographic Databases

Juliano Lopes de Oliveira
Claudia Bauzer Medeiros

Relatório Técnico DCC–95-08

Junho de 1995

A Direct Manipulation User Interface for Querying Geographic Databases

Juliano Lopes de Oliveira **Claudia Bauzer Medeiros**

Abstract

This paper presents an architecture for a direct manipulation user interface for browsing and querying geographic data. This interface provides users with a high level object oriented conceptual view of the underlying database, independent of the database's native data model. It lets users manipulate different representations of a single georeferenced entity, thereby adding a new degree of flexibility to querying facilities.

1 Introduction

Geographic Information Systems – GIS – are systems that perform data management and retrieval operations for *georeferenced data*. This term refers to data about geographic phenomena associated with their physical location (coordinates) and spatial relationships [Car89]. Examples of GIS applications are urban planning, thematic and statistic mapping for natural resource management and public utility management [Aro89].

Data handled by GIS are often stored in geographic database systems, which are based on combining a (usually) relational DBMS with special handlers which manipulate specific aspects of georeferenced data.

Once stored into a GIS, georeferenced data can be classified into two main categories [VE93]:

- *aspatial data* – traditional, descriptive alphanumeric attributes, handled by conventional DBMS;
- *spatial data* – attributes that describe the geometry and location of geographic phenomena. Spatial data has geometrical (e.g., size) and topological (e.g., connectivity) properties.

Geographic applications rely heavily on sophisticated graphical displays, usually in some cartographic form. GIS users must therefore be able to define the *presentation* (i.e., display characteristics) of the answer to queries. Normally, this is supported in a GIS by processing the query in three steps:

1. The query processing manager retrieves from the database the data that the user has selected in the query. This initial step combines traditional retrieval with specific spatial processing algorithms (e.g., from computational geometry);

2. (optional) The retrieved data is combined through data analysis functions to obtain derived information and correlations among the data;

3. The retrieved (or derived) data are displayed according to the user's presentation specifications (symbols, texture, color).

Throughout this process, users are expected to know the organization of the underlying data (corresponding to the database schema) and available data ranges. Presentation specification often requires learning additional concepts.

This paper presents a solution to the browsing and retrieval issue, by adding a new degree of flexibility to the database query interface. It consists of an architecture that offers to the user an object-oriented direct manipulation interface that allows accessing data in two different levels:

- Meta-data level: it helps users navigate through the schema of the geographic database, selecting the types of data for subsequent querying; and
- Data level: the user defines textual and visual query predicates, through simultaneous use of two interaction paradigms:
  - textual (traditional); and
  - visual - in this second interaction mode, the user can at the same time determine the value ranges for aspatial attributes, as well as the display characteristics of both conventional and spatial data.

This architecture has already been implemented, for traditional databases, in the GOODIES tool [OA93a].

The remainder of this paper is organized as follows. Section 2 discusses some issues in the development of GIS applications from a database point of view, pointing out problems in the field of visualization and query interfaces. Section 3 presents the architecture of the proposed interface. Section 4 describes a working session using this architecture. Finally, section 5 presents conclusions and directions for future work.

2 GIS applications and databases

The database community has contributed to GIS research by developing data structures and algorithms in two fields: *spatial data structures* (e.g., quadtrees) which allow the efficient manipulation of objects in 2D and 3D space; and structures for supporting *geometric operations*.

In geographic query processing, the emphasis has been on computation (i.e., translating the user's requests into appropriate database operations) and optimization (by indexing data with spatial structures, and using computational geometry algorithms). It is only recently, as pointed out by [VE93], that the interaction aspect has been considered in the design of GIS interfaces.

Users' modelling of georeferenced entities is associated with different perceptions of the world: the *field* model and the *object* model [FG90, Goo91]. The *object* view treats the world as a surface littered with recognizable objects, which exist independent of any definition (e.g., a given road). In this model, two objects can occupy the same place (e.g., a turnpike in the road). Database entities correspond to these recognizable objects.

Field data is processed in tesseral format (spatial objects described as polygonal units of space – cells – in a matrix). Each cell contains one thematic value. Cells may have different shapes; square cells are called *pixels*. The *raster* format (which is often used as the generic name for tesseral data) is just one special type of tesselation with rectangular grid format.

Object data is stored as points, lines and polygons (the *vector* format model), using lists of coordinate pairs. This type of format is usually more adequate for representing man-made artifacts (e.g., bridges), whereas the field model is adopted mostly in environmental applications [Cou92]. One challenge, for the interface designer, is to accommodate both types of model in query processing mechanisms.

Geographic reality is perceived according to the user needs and the application domain. This often requires that different *representations* be stored (e.g, varying according to time or scale), incurring thereby into problems such as establishing links and consistency among distinct representations of a given region. Queries in geographic databases must thus combine and integrate both spatial and aspatial data, as well as consider the different representations for a given georeferenced entity.

GIS interfaces usually support two types of interaction mechanisms [MP94]:

- textual query languages; and
- interactive manipulation of geographic elements

Most textual query languages are based on extensions of relational query languages such as SQL (e.g., [Gut88, Goh89, PZ91, Ooi90, Lor91]). The disadvantages of this approach are pointed out in [Ege92]. Other textual query mechanisms usually rely on extending object oriented query languages with spatial processing methods (e.g., [SV92]). Some recent results exploit the use of natural language in queries, for restricted application domains (e.g., [Wan94]).

Some GIS querying mechanisms allow specifying the output visual characteristics as part of a textual query (see [Ege94]). Other mechanisms allow determining multiple presentations of a given georeferenced data set, as well as defining the amount of objects that are to be displayed for a given window size (e.g., [AKK94]). Most commercial systems allow predefinition of the presentation characteristics of stored elements by clicking on menus which allow associating each data type with its visualization features (color, texture).

Visual query languages for database systems allow the user to associate weights to values in the database, which are displayed in some graphical form. Users can infer relationships among data from this visual output (e.g., by differences in color, or from relative distances – [Spo93, KK94]). However, these languages are not directly applicable to geographic databases. Very few proposals exist for visual languages for GIS. These rely on letting the user specify predicates by combining a predefined set of spatial operators available iconically (e.g., [CCM92, VMS+93]). For some application domains, queries may be specified as directed graphs with predicates (e.g., [Men93]), hypertext facilities (e.g., [LR93]) or combining forms and icons [WCY89]. The problems in visual queries consist mainly in providing the query translation mechanism with enough information to disambiguate the users' requests, since some of the iconic operators are semantically overloaded (e.g., [BM92, DC91]).

The design of interfaces for geographic databases, and, more specifically, GIS, must thus necessarily take at least the following aspects into consideration:

- Integration between the interface and the underlying database

  GIS users are usually experts in a given application domain, but know nothing about the underlying data management system. Users must not be forced to learn more than one query language, nor have to know the underlying database schema.

- Representation of spatial concepts

  An interface must support multiple representations of a given georeferenced entity, depending on the users' needs [Rig95].

- Query types

  Basic types of queries investigate contents (describing contents of a region), location (determining location of elements satisfying a given set of predicates), time/trends (what has changed in an area through time, and how), alternative evaluation, routing. An interface must thus be able to provide users with enough tools to access data in all these ways.

The next section describes our architecture for a browsing and querying interface for geographic databases, which combines both textual and visual characteristics. It contemplates the above issues, thereby adding a new degree of flexibility to geographic databases.

3 System Architecture

There are two basic approaches to integrate a user interface system to a DBMS, and in particular to a GIS: *strong integration* and *weak integration* [Voi94]. In the former, the user interface is part of the geographical system, sharing its data model and taking advantage of the knowledge about the internal data structures. As a consequence there is great difficulty in using data from different sources, and it is not possible to adapt the same user interface to different GIS. In the second approach, weak integration, the user interface is considered an external module, and is therefore adaptable to more than one system. However, it presents the disadvantage of demanding the definition of communication and data conversion protocols between the user interface system and the geographical system.

The architecture presented here is based on the weak integration approach, since its advantages outnumber its shortcomings. First, there is a world-wide trend towards the development of open systems, which can be integrated with other products, and GIS are among the systems that pursue this kind of architecture (e.g., [AYA+92, VvO92, GR93, PMP93, VS94]). Second, this kind of integration provides independence and improves specialization of functionality of each component (user interface and geographical data management). Furthermore, this philosophy allows importing specialized packages, such as graphical libraries, into the user interface. Last but not least, this integration approach allows the progressive inclusion of new services and functions in the GIS.

In the weak integration approach, the user interface can be coupled to different GIS. This approach needs therefore a rich data model to represent concepts used in different GIS. We chose the object-oriented GIS model of [CFS+94]. This data model suports both the field and the object views of the geographical world. It consists of four levels of abstraction: the real world level (the real geographical phenomena); the conceptual level (an abstract view of these phenomena, in which operations are independent from the representation of the data); the representation level (where operations are specialized to each particular representation of a geographical entity); and the physical level (which deals with issues that provide efficient storage and retrieval of data, for instance, spatial indexes and access methods). Georeferenced entities are conceptually classified into geo-objects (object view) and geo-fields (field view). Each such class has its own high-level operations and particular representations. The model supports the multiple representation paradigm, in which a given georeferenced phenomenon may be perceived differently according to application needs. By supporting this model, the interface architecture establishes a framework for two important end-users requirements: users can manipulate multiple representations at the interface level; and users can define distinct presentations (display characteristics) for each representation.

The architecture of the interface, presented in figure 1, has three main components:

- The *user dialog* module manages the user interaction with the interface system. It is responsible for two main tasks: the creation and management of presentations, and the translation of user's requests into operations of the underlying geographic database model (and vice-versa). These tasks are performed, respectively, by the presentation manager and by the interaction manager. The binding with the graphic toolkit is another important task within the user dialog module.

  The definition and management of presentations are handled, in the screen, through two areas: a control area for query definition and a display area for result visualization. This task involves graphical display operations (display area) and dynamic construction of widgets (control area). Each representation of the data can be visualized through different presentations (e.g., graphical, diagrams or tabular).

  The second main task is the translation of direct manipulation actions of the user into high-level conceptual operations on georeferenced data, managed by the data model module.

- The *data model* module is responsible for providing to the user a view of the underlying database which is compatible with the adopted data model [CFS+94]. The conceptual manager is responsible for the object-oriented schema that describes geographical entities. The representation manager records the different representations associated with each conceptual entity. The main task of the data model module is to support a high level conceptual view of data. It therefore supports browsing on concepts rather than on representations, allowing the user to see multiple representations of the same

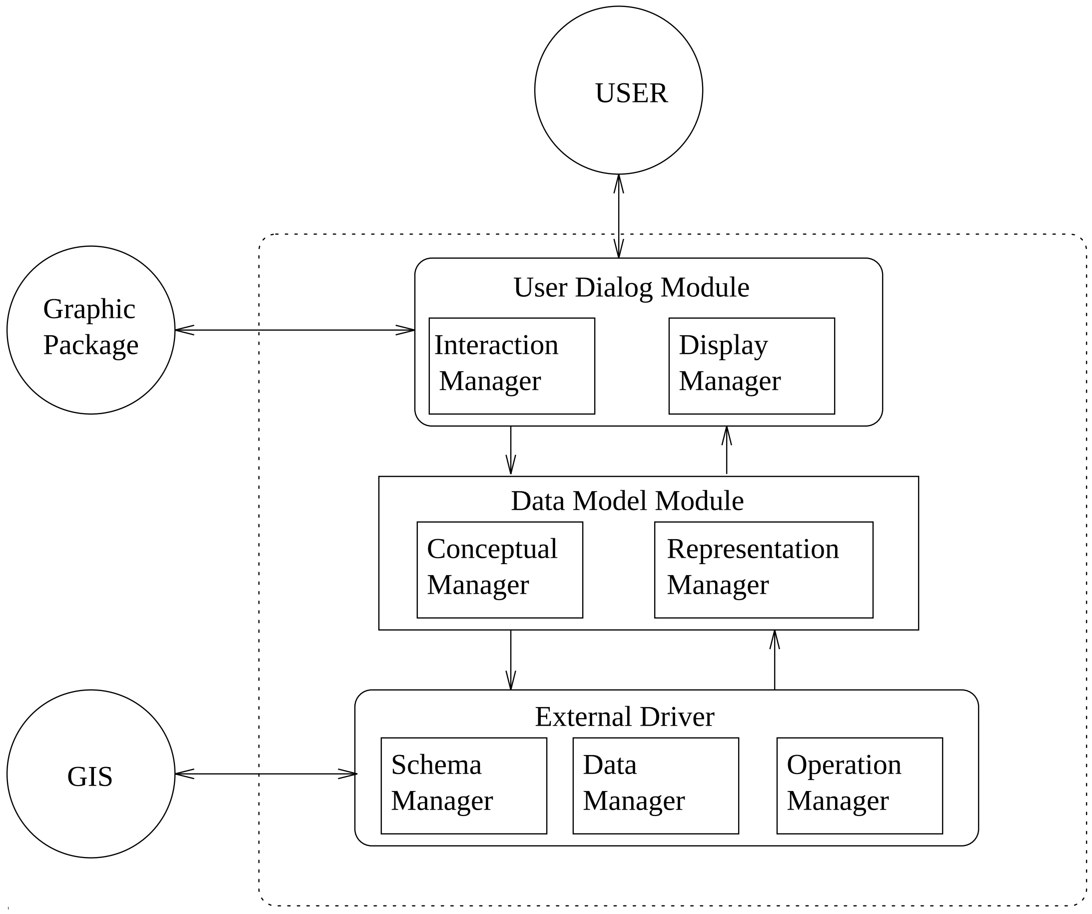

Figure 1: Architecture of the User Interface System

data. Another important task of this module is to convert conceptual operations into representation dependent operations, which are sent to the external driver.

- The *external driver* converts data from the format used in the GIS to the internal data model of the interface and vice-versa. This is achieved by means of a communication protocol that is based on primitive operations that allow retrieving from the GIS database schemas, class descriptions and data values. The approach used in the definition of this module is the same we used to integrate a user interface to different object-oriented database systems [OA93b]. The interface sends queries to the GIS using the primitive operations (Get-Schema, Get-Class, and Get-Value), and the external driver implements these operations according to the syntax of the underlying geographic DBMS.

Although figure 1 shows only one external driver and one GIS, the architecture is perfectly able to deal with many underlying GIS, depending on adding new external driver modules. The architecture of the interface system supports distinct conceptual views of the geographical space. Each conceptual view corresponds to an object-oriented database schema built from the underlying GIS schemas according to the definition of the conceptual level of the data model. To provide facilities for navigating in both object and schema level, the user dialog module uses the GOODIES open system [OA93a], a generic OODBMS browser.

4 A Query Interface for Objects and Fields

The previous section showed an architecture for a weekly coupled GIS user interface. The architecture has three main components which handle, respectively, the communication with the geographic database, the interface data model and the interaction with the user. The external driver module is an extension of our previous work in the integration of user interfaces in OODBMS [OA93b]. The data model module relies on the conceptual and representation levels of the four-level object-oriented data model of [CFS+94]. Finally the user dialog module is responsible for translating user actions into operations on georeferenced data and for converting the data provided through the data model module into windows presented to the user. This section presents how the interface guides the user through browsing and querying georeferenced data.

4.1 Working Session – Guiding the User through Data Selection

One important problem in GIS interfaces is how to support user queries without requiring specialized knowledge of the underlying database. This is solved in the present interface by helping the user through a sequence of steps that will construct the query. Once the data is retrieved and displayed, the user can interact with it through direct manipulation mechanisms [Shn83].

The steps that are provided are based on our experience of working with different GIS, both for environmental control and utility management applications. The steps are divided into *schema definition* of the query and query *predicate specification*. Data is thus retrieved into a pre-established query schema similar to a view mechanism [MM91]. The schema definition steps are:

1. schema navigation and class selection;
2. choice of class entities;
3. definition of representations for each selected entity;
4. determination of visual aspect.

When the user starts a session, the external driver establishes a connection with the underlying geographic database and retrieves the available schemas using the primitive operation Get-Schema. In general, a database contains data from a particular project or application, and refers to the same geographical region. Using these data the conceptual manager mounts the schemas in the interface conceptual model (geo-schemas).

The user can browse through different geo-schemas to select the elements (classes) that are relevant to the user's needs. As the elements are selected, they are added to the control area of the interface. The selection of representation and visualization characteristics for entities is a trivial process of selecting options in a menu. The same georeferenced entity may have more than one representation. A geo-field, for example, may be represented by a satellite image or by a set of coordinate value pairs. Thus, after choosing an entity, the user can select the desired representation for the entity. The interface system also allows the user to select the *look* of the entity in the display, i.e, it associates different kinds of presentations to each representation. For instance, a geo-field that is represented by pairs (coordinate, value) may be presented in a tabular form, as a graph (histogram, pie-chart, and others) or as a map (e.g., a 2D surface).

Geo-fields are represented by a *slider* labeled with the name of the class in the conceptual schema. The slider shows three values: the minimum and maximum values that the geo-field may assume, and a current value selected by direct manipulation. Geo-objects are represented by a push button labeled with the name of the geo-object in the conceptual schema. Clicking the button causes the display of an auxiliary window containing the geo-object attributes. Once the schema is defined, the user formulates a query by selecting ranges for geo-fields, values for attributes of geo-fields, and typing alphanumerical predicates. These predicates can combine comparison operators, logical connectors and spatial functions implemented in the underlying GIS.

4.2 Requirements and Design

The characteristics of a user interface system for GIS impose several challenges to the interface developer. The main requirements for the interface presented here were: to be independent from an specific GIS; to provide powerful tools for expert users; to reduce the time needed for novice users to use the querying facilities; to improve the process of selection and exploration of data; to provide a conceptual view of the geographical space in terms of objects and fields; to support not only data level but also schema level navigation; to allow multiple representations of the same geographic entity, and multiple visualizations of a single data representation.

Some of the requirements were met by the architecture specification. Independence from a particular GIS was ensured by the week integration approach. The conceptual view and representation facilities were ensured by basing the architecture on the appropriate data model. The other requirements were provided by supporting a stepwise query definition and by the characteristics of the display manager.

In order to meet the usability requirements, the user dialog module implements a variation of the *dynamic query* paradigm [AWS92]. This paradigm is well suited for data sets with multiple search keys, and is based on the graphical representation of both queries and results, on the use of direct manipulation of graphical objects (widgets) and on the immediate feedback for this kind of interaction. With the recent advances in software and hardware, the dynamic query paradigm has gained interest, because it is now possible to provide the necessary (almost) real-time answers for complex queries. Interface systems using this paradigm are beginning to appear in the literature [KK94, AKK94]. With dynamic queries, the interface system is usually very easy to use, even for novice users; at the same time, the paradigm can provide powerful tools for expert users. But the most desired feature of dynamic queries is that it encourages users to explore the potential of informations in the database.

4.3 Using the interface: a typical example

The features of the interface can be better understood by means of an example. Consider a civil defense application where the user is interested in identifying risk areas for natural catastrophes, for the region stored in the database. Consider a query concerning spatial distribution of cities as regards forests and declivities. The user identifies that there is a potential fire risk for all cities that are close to coniferous forests in high declivity areas, and wants to visualize these data.

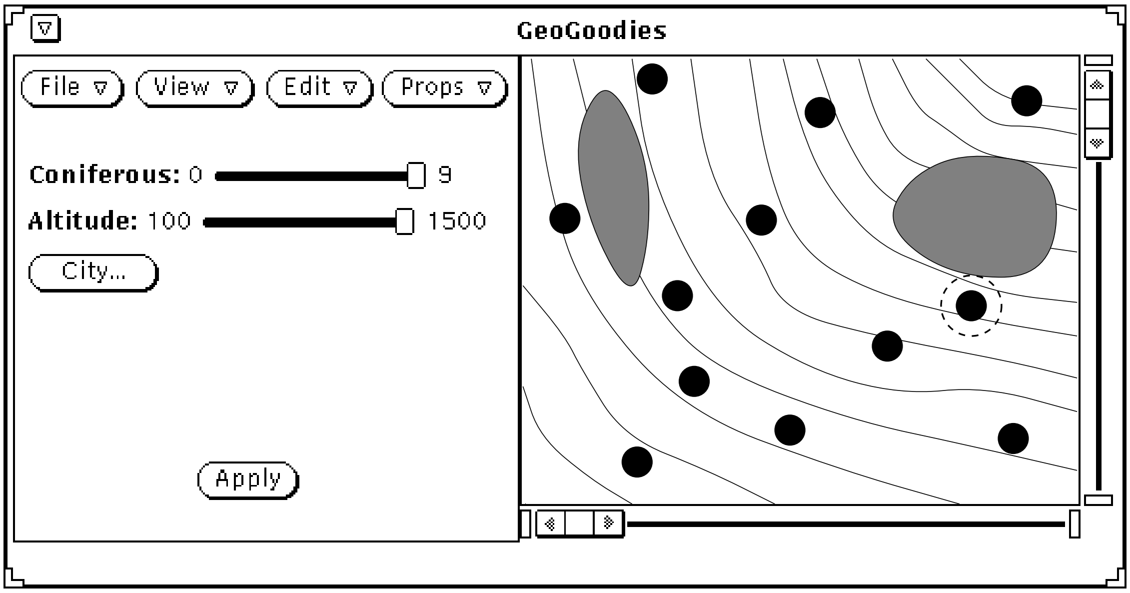

Figure 2: Final scenario with selection of geo-object

First the user browses through the object-oriented schemas, looking for classes that are related to the defined problem. During this process the user gains knowledge about all the information available in the underlying GIS, in a very abstract level. The user finds the entity *Vegetation*, which is the root of a class hierarchy containing subclass *Coniferous*, which is selected for querying. In response to this selection, the interface system indicates the only representation available in the database for the entity, and offers the user the presentation options that are allowed for this kind of data representation. The user selects the default presentation: a color planar map.

Continuing the schema level browsing, the user realizes that there is no schema with a *Declivity* class, but there is a tightly related class, *Altitude*, which is thus selected. The interface system finds out two representations for the selected class in the GIS: triples of coordinates (x, y, z) and satellite images, both of which correspond to field views. The user selects the first representation, and chooses the *isoline* presentation option (to see the

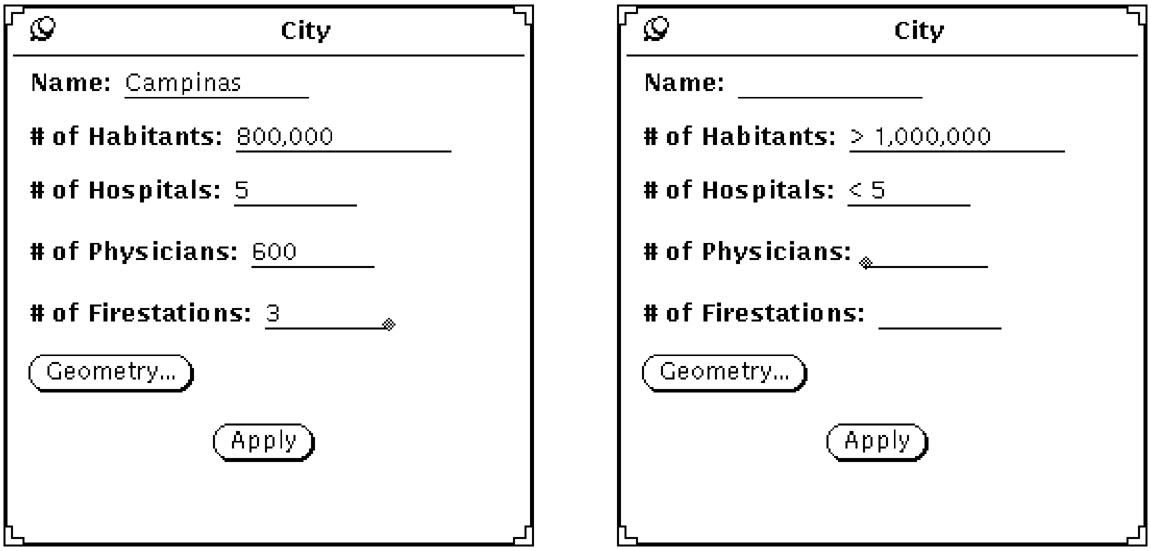

Figure 3: Visualizing and querying attributes of a geo-object

declivity aspect).

The last concept involved in the user task are urban areas, and the user selects the class *City*, with the respective representation and presentation options. If no predicates are indicated, all data elements are presented. The final result is displayed in figure 2, where the contents of the three classes are overlayed and displayed in one scenario. The legend for each class can be visualized through the *Legend* option of the *View* menu. The sliders indicate that the user is getting the complete range of altitudes (100 to 1500 meters) and all the nine different types of coniferous forest stored in the database. Moving these sliders allow users to specify the range of values for visualization of such geo-fields.

Examining the control area of the main window, the user can infer that the *City* class is a geo-object. Clicking the *City* button causes the interface system to present an auxiliary window containing all the *City* class attributes. The user is allowed to select relevant attributes, discarding all others, as well as defining textual predicates for them. Values of attributes of given objects can be visualized through selecting (clicking) them in the graphical area of the main window. Figure 2 shows the selection of an object (which is highlighted) close to a coniferous forest, and figure 3 (left) shows the object's attribute window.

The user can apply predicates, as defined in section 4.1, to restrict the displayed objects. For instance, figure 3 (right) shows a user-defined predicate that selects for display the cities with more than one million habitants which have less than five hospitals. The result of the query is shown in figure 4.

Different geographic queries may be posed by applying the corresponding functions available in the underlying database (e.g., distance, area). These functions are activated by a combination of selection of the appropriate geo-classes and clicking the function (in the

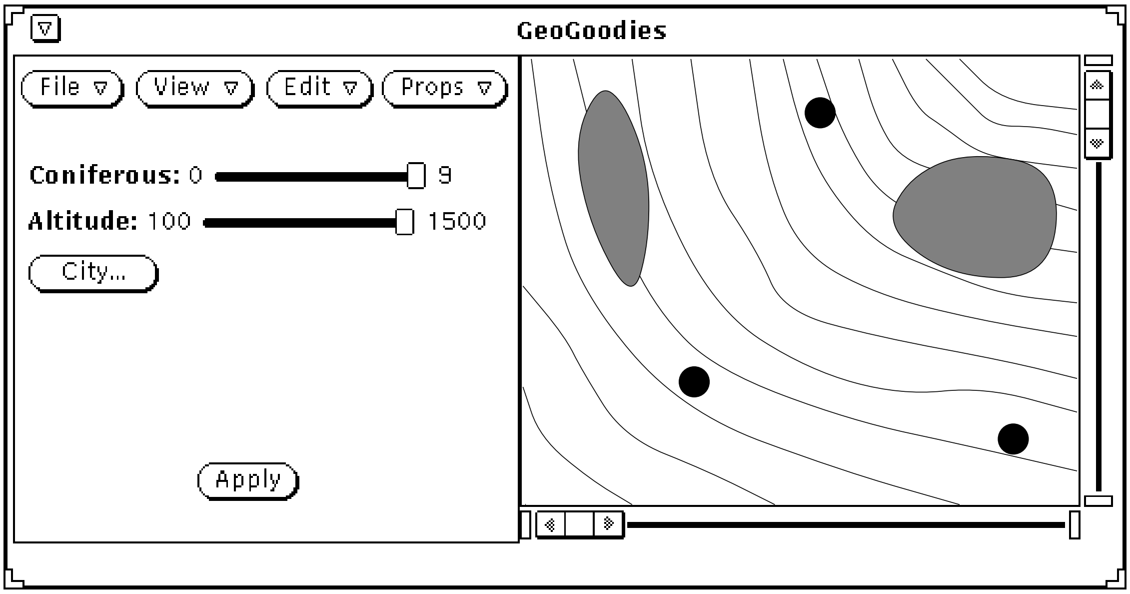

Figure 4: Query Results in Main Window

Edit menu of the main window).

5 Conclusions and future work

This paper presented an architecture of a direct manipulation user interface for geographic databases. This architecture has the following advantages over other GIS interfaces:

- it does not require that users have previous knowledge of the underlying database schema or query language;
- it gives the user a high level object-oriented view of the underlying data, which approaches the user to a conceptual view of the world;
- it supports manipulation of different representations of data, and distinct presentations thereof;
- it allows users to pose queries by mixing textual and visual paradigms, thereby helping determine display characteristics at the same time query predicates are posed

This user interface architecture was implemented, for traditional databases, in the GOODIES tool [OA93a]. This is now being extended to accommodate queries on geographic databases.

Other extensions being considered concern the architecture's design. An important issue is to try to support updates through the interface, which necessarily will need different translations depending on the representation and presentation chosen for a given data. This, in turn, falls into the problem of spatial integrity constraints, itself an open issue. Another problem being considered is the maintenance of the spatial functions provided through the underlying database.

Acknowledgements

The work described in this paper was partially financed by grants from FAPESP and CNPq, as well as grants from CNPq Protem project GEOTEC and the European Community ITDC project GEOTOOLS number 116-82152.

References

[AKK94] M. Arikawa, H. Kawakita, and Y. Kambayashi. Dynamic Maps as Composite Views of Varied Geographic Database Servers. In *Proceedings of the International Conference on Applications of Databases*, 1994.

[Aro89] S. Aronoff. *Geographic Information Systems*. WDL Publications, Canada, 1989.

[AWS92] C. Ahlberg, C. Williamson, and B. Shneiderman. Dynamic Queries for Information Exploration: An Implementation and Evaluation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, May 1992.

[AYA+92] D. Abel, S. Yap, R. Ackland, M. Cameron, D. Smith, and G. Walker. Environmental decision support system project: an exploration of alternative architectures for geographical information systems. *International Journal of Geographical Information Systems*, 6(3):193–204, 1992.

[BM92] P Boursier and M Mainguenaud. Spatial Query Languages: Extended SQL vs Visual Languages vs Hypermaps . In *Proc 5th International Symposium on Spatial Data Handling*, pages 249–259, 1992. Volume 1.

[Car89] J. Carter. *Fundamentals of Geographic Information Systems: A Compendium*, chapter On Defining the Geographic Information System, pages 3–8. American Society for Photogrammetry and Remote Sensing, 1989.

[CCM92] M. Consens, I. Cruz, and A. Mendelzon. Visualizing Queries and Querying Visualization. *ACM Sigmod Record*, 21(1):39–46, 1992.

[CFS+94] G. Câmara, U. Freitas, R. Souza, M. Casanova, A. Hemerly, and C. Medeiros. A model to cultivate objects and manipulate fields. In *Proceedings of the second ACM Workshop on Advances in GIS*, pages 30–37, Gaithersburg, Maryland, USA, December 1994.

[Cou92] H. Couclelis. People Manipulate Objects (but Cultivate Fields): Beyond the Raster-Vector Debate in GIS. In *Proc International Conference on GIS - From Space to Territory: Theories and Methods of Spatial Reasoning*, Springer Verlag Lecture Notes in Computer Science 639, pages 65–77, 1992.

[DC91] M. Maingenaud D. Calcinelli. The Management of the Ambiguities in a Graphical Query Language for GIS. In *Proc. 2nd Symposium Spatial Database Systems*, pages 99–118. Spring Verlag Lecture Notes in Computer Science 525, 1991.

[Ege92] M. Egenhofer. Why not SQL! *International Journal of Geographical Information Systems*, 6(2):71–86, 1992.

[Ege94] M. Egenhofer. Spatial SQL: A Query and Presentation Language. *IEEE Transactions on Knowledge and Data Engineering*, 6(1):86–95, 1994.

[FG90] A. Frank and M. Goodchild. Two Perspectives on Geographical Data Modelling. Technical Report 90-11, National Center for Geographic Information and Analysis, 1990.

[Goh89] P-C. Goh. A Graphic Query Language for Cartographic and Land Information Systems. *International Journal of Geographical Information Systems*, 3(3):245–256, 1989.

[Goo91] M. Goodchild. Integrating GIS and Environmental Modeling at Global Scales. In *Proc GIS/LIS'91*, volume 1, pages 117–127, 1991.

[GR93] O. Günter and W. Riekert. The design of godot: an object-oriented geographic information system. *Bulletin of the Technical Committee on Data Engineering*, 16(3):4–9, September 1993. Special Issue on Geographical Information Systems.

[Gut88] R. Guting. Geo-relational Algebra: a Model and Query Language for Geometric Database Systems. In *Proc. 1st EDBT Conference*, pages 506–527, 1988.

[KK94] D. Keim and H.-P. Kriegel. VisDB: Database Exploration using Multidimensional Visualization. *IEEE Computer Graphics and Applications*, 14(5):40–49, 1994.

[Lor91] R. Lorie. The Use of a Complex Object Language in Geographic Data Management. In *Proc. 2nd Symposium Spatial Database Systems*, pages 319–337. Springer Verlag Lecture Notes in Computer Science 525, 1991.

[LR93] T. Linsey and J. Raper. HyperArc: a Task-oriented Hypertext GIS Interface. *International Journal of Geographical Information Systems*, 7(5):435–452, 1993.

[Men93] A. Mendelzon et al. Declarative Database Visualization: Recent Papers from the Hy+/GraphLog Project . Technical Report CSRI-285, University of Toronto, 1993.

[MM91] J. Mamou and C. B. Medeiros. Interactive Manipulation of Object-Oriented Views. In *IEEE Proceedings of the International Conference on Data Engineering*, pages 60–69, Kobe, Japan, April 1991.

[MP94] C. B. Medeiros and F. Pires. Databases For GIS. *ACM SIGMOD Record*, 1994.

[OA93a] J. L. Oliveira and R. O. Anido. Browsing and Querying in Object Oriented Databases. In *Second International Conference on Information and Knowledge Management*, pages 364–373, Washington, DC, USA, November 1993.

[OA93b] J. L. Oliveira and R. O. Anido. Integration of a Browsing Interface in Different Object-Oriented Database Systems. In *Thirteenth Brasilian Computer Society Congress*, pages 61–75, Florianópolis, SC, Brazil, September 1993. Written in portuguese.

[Ooi90] B. C. Ooi. *Efficient Query Processing in Geographic Information Systems*. Springer Verlag Lecture Notes in Computer Science, 1990.

[PMP93] N. Pissinou, K. Makki, and E. Park. Towards the Design and Development of a New Architecture for Geographic Information Systems . In *Proc. 2nd International Conference on Information and Knowledge Management*, pages 565–573, 1993.

[PZ91] P.Svensson and Z.Huang. Geo-Sal: A query Language for Spatial Data Analysis. In *Proc. 2nd Symposium Spatial Database Systems*, pages 119–140. Springer Verlag Lecture Notes in Computer Science 525, 1991.

[Rig95] P. Rigaux. *Interfaces Graphiques pour Bases de Données Spatiales: Application à la Représentation Multiple*. PhD thesis, CEDRIC - Conservatoire National des Arts et Metiers, 1995.

[Shn83] B. Shneiderman. Direct Manipulation: A Step Beyond Programming Languages. *IEEE Computer*, 16(8):57–69, August 1983.

[Spo93] A. Spoerri. InfoCrystal: A visual tool for information retrieval and management. In *Proceedings, CIKM Conference*, pages 11–20, 1993.

[SV92] M. Scholl and A. Voisard. *Building and Object-oriented System – the Story of O2*, chapter Geographic Applications – an Experience with O2. Morgan Kaufmann, California, 1992.

[VE93] G. Volta and M. Egenhofer. Interaction with GIS Attribute Data Based on Categorical Coverage. In *European Conference Spatial Information Theory - COSIT*, pages 215–232, 1993.

[VMS+93] K. Vadaparty, R. Molmen, H. Salem, P. Whiting, and S. Naqvi. Towards a Fully Sheetless GIS with Incremental Querying. In *Proc. ACM/ISCA Workshop on Advances in Geographic Information Systems*, pages 94–99, 1993.

[Voi94] A. Voisard. Designing and Integrating User Interfaces of Geographic Database Applications. In *Proceedings of the 1994 ACM Workshop on Advanced Visual Interfaces*, pages 133–142, 1994.

[VS94] A. Voisard and H. Schweppe. A Multilayer Approach to the Open GIS Design Problem. In *Proceedings of the second ACM Workshop on Advances in GIS*, pages 23–29, Gaithersburg, Maryland, USA, December 1994.

[VvO92] T. Vijlbrief and P. van Oosterom. The GEO++ system: an extensible GIS. In *Proceedings of the Symposium on Spatial Data Handling*, pages 40–50, August 1992.

[Wan94] F. Wang. Towards a Natural Language User Interface: an Approach of fuzzy Query. *International Journal of Geographical Information Systems*, 8(2):143–162, 1994.

[WCY89] J. Wu, T. Chen, and L. Yang. QPF: a Versatile Query Language for a Knowledge-based Geographical Information System. *International Journal of Geographical Information Systems*, 3(1):51–58, 1989.

Relatórios Técnicos – 1992

92-01 **Applications of Finite Automata Representing Large Vocabularies,** *C. L. Lucchesi, T. Kowaltowski*

92-02 **Point Set Pattern Matching in d-Dimensions,** *P. J. de Rezende, D. T. Lee*

92-03 **On the Irrelevance of Edge Orientations on the Acyclic Directed Two Disjoint Paths Problem,** *C. L. Lucchesi, M. C. M. T. Giglio*

92-04 **A Note on Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams,** *W. Jacometti*

92-05 **An (l, u)-Transversal Theorem for Bipartite Graphs,** *C. L. Lucchesi, D. H. Younger*

92-06 **Implementing Integrity Control in Active Databases,** *C. B. Medeiros, M. J. Andrade*

92-07 **New Experimental Results For Bipartite Matching,** *J. C. Setubal*

92-08 **Maintaining Integrity Constraints across Versions in a Database,** *C. B. Medeiros, G. Jomier, W. Cellary*

92-09 **On Clique-Complete Graphs,** *C. L. Lucchesi, C. P. Mello, J. L. Szwarcfiter*

92-10 **Examples of Informal but Rigorous Correctness Proofs for Tree Traversing Algorithms,** *T. Kowaltowski*

92-11 **Debugging Aids for Statechart-Based Systems,** *V. G. S. Elias, H. Liesenberg*

92-12 **Browsing and Querying in Object-Oriented Databases,** *J. L. de Oliveira, R. de O. Anido*

Relatórios Técnicos – 1993

93-01 **Transforming Statecharts into Reactive Systems,** *Antonio G. Figueiredo Filho, Hans K. E. Liesenberg*

93-02 **The Hierarchical Ring Protocol: An Efficient Scheme for Reading Replicated Data,** *Nabor das C. Mendonça, Ricardo de O. Anido*

93-03 **Matching Algorithms for Bipartite Graphs,** *Herbert A. Baier Saip, Cláudio L. Lucchesi*

93-04 **A lexBFS Algorithm for Proper Interval Graph Recognition,** *Celina M. H. de Figueiredo, João Meidanis, Célia P. de Mello*

93-05 **Sistema Gerenciador de Processamento Cooperativo,** *Ivonne. M. Carrazana, Nelson. C. Machado, Célio. C. Guimarães*

93-06 **Implementação de um Banco de Dados Relacional Dotado de uma Interface Cooperativa,** *Nascif A. Abousalh Neto, Ariadne M. B. R. Carvalho*

93-07 **Estadogramas no Desenvolvimento de Interfaces,** *Fábio N. de Lucena, Hans K. E. Liesenberg*

93-08 **Introspection and Projection in Reasoning about Other Agents,** *Jacques Wainer*

93-09 **Codificação de Seqüências de Imagens com Quantização Vetorial,** *Carlos Antonio Reinaldo Costa, Paulo Lício de Geus*

93-10 **Minimização do Consumo de Energia em um Sistema para Aquisição de Dados Controlado por Microcomputador,** *Paulo Cesar Centoducatte, Nelson Castro Machado*

93-11 **An Implementation Structure for RM-OSI/ISO Transaction Processing Application Contexts,** *Flávio Morais de Assis Silva, Edmundo Roberto Mauro Madeira*

93-12 **Boole's conditions of possible experience and reasoning under uncertainty,** *Pierre Hansen, Brigitte Jaumard, Marcus Poggi de Aragão*

93-13 **Modelling Geographic Information Systems using an Object Oriented Framework,** *Fatima Pires, Claudia Bauzer Medeiros, Ardemiris Barros Silva*

93-14 **Managing Time in Object-Oriented Databases,** *Lincoln M. Oliveira, Claudia Bauzer Medeiros*

93-15 **Using Extended Hierarchical Quorum Consensus to Control Replicated Data: from Traditional Voting to Logical Structures,** *Nabor das Chagas Mendonça, Ricardo de Oliveira Anido*

93-16 $\mathcal{LL}$ – **An Object Oriented Library Language Reference Manual,** *Tomasz Kowaltowski, Evandro Bacarin*

93-17 **Metodologias para Conversão de Esquemas em Sistemas de Bancos de Dados Heterogêneos,** *Ronaldo Lopes de Oliveira, Geovane Cayres Magalhães*

93-18 **Rule Application in GIS – a Case Study,** *Claudia Bauzer Medeiros, Geovane Cayres Magalhães*

93-19 **Modelamento, Simulação e Síntese com VHDL,** *Carlos Geraldo Krüger e Mário Lúcio Côrtes*

93-20 **Reflections on Using Statecharts to Capture Human-Computer Interface Behaviour,** *Fábio Nogueira de Lucena e Hans Liesenberg*

93-21 **Applications of Finite Automata in Debugging Natural Language Vocabularies,** *Tomasz Kowaltowski, Cláudio Leonardo Lucchesi e Jorge Stolfi*

93-22 **Minimization of Binary Automata,** *Tomasz Kowaltowski, Cláudio Leonardo Lucchesi e Jorge Stolfi*

93-23 **Rethinking the DNA Fragment Assembly Problem,** *João Meidanis*

93-24 EGOLib — **Uma Biblioteca Orientada a Objetos Gráficos,** *Eduardo Aguiar Patrocínio, Pedro Jussieu de Rezende*

93-25 **Compreensão de Algoritmos através de Ambientes Dedicados a Animação,** *Rackel Valadares Amorim, Pedro Jussieu de Rezende*

93-26 GeoLab: **An Environment for Development of Algorithms in Computational Geometry,** *Pedro Jussieu de Rezende, Welson R. Jacometti*

93-27 **A Unified Characterization of Chordal, Interval, Indifference and Other Classes of Graphs,** *João Meidanis*

93-28 **Programming Dialogue Control of User Interfaces Using Statecharts,** *Fábio Nogueira de Lucena e Hans Liesenberg*

93-29 EGOLib – **Manual de Referência,** *Eduardo Aguiar Patrocínio e Pedro Jussieu de Rezende*

Relatórios Técnicos – 1994

94-01 **A Statechart Engine to Support Implementations of Complex Behaviour,** *Fábio Nogueira de Lucena, Hans K. E. Liesenberg*

94-02 **Incorporação do Tempo em um SGBD Orientado a Objetos,** *Ângelo Roncalli Alencar Brayner, Claudia Bauzer Medeiros*

94-03 **O Algoritmo KMP através de Autômatos,** *Marcus Vinícius A. Andrade e Cláudio L. Lucchesi*

94-04 **On Edge-Colouring Indifference Graphs,** *Celina M. H. de Figueiredo, João Meidanis, Célia Picinin de Mello*

94-05 **Using Versions in GIS,** *Claudia Bauzer Medeiros and Geneviève Jomier*

94-06 **Times Assíncronos: Uma Nova Técnica para o Flow Shop Problem,** *Hélvio Pereira Peixoto e Pedro Sérgio de Souza*

94-07 **Interfaces Homem-Computador: Uma Primeira Introdução,** *Fábio Nogueira de Lucena e Hans K. E. Liesenberg*

94-08 **Reasoning about another agent through empathy,** *Jacques Wainer*

94-09 **A Prolog morphological analyser for Portuguese,** *Jacques Wainer, Alexandre Farcic*

94-10 **Introdução aos Estadogramas,** *Fábio N. de Lucena, Hans K. E. Liesenberg*

94-11 **Matching Covered Graphs and Subdivisions of K_4 and $\overline{C_6}$,** *Marcelo H. de Carvalho and Cláudio L. Lucchesi*

94-12 **Uma Metodologia de Especificação de Times Assíncronos,** *Hélvio Pereira Peixoto, Pedro Sérgio de Souza*

Relatórios Técnicos – 1995

95-01 **Paradigmas de algoritmos na solução de problemas de busca multidimensional,** *Pedro J. de Rezende, Renato Fileto*

95-02 **Adaptive enumeration of implicit surfaces with affine arithmetic,** *Luiz Henrique de Figueiredo, Jorge Stolfi*

95-03 **W3 no Ensino de Graduação?,** *Hans Liesenberg*

95-04 **A greedy method for edge-colouring odd maximum degree doubly chordal graphs,** *Celina M. H. de Figueiredo, João Meidanis, Célia Picinin de Mello*

95-05 **Protocols for Maintaining Consistency of Replicated Data,** *Ricardo Anido, N. C. Mendonça*

95-06 **Guaranteeing Full Fault Coverage for UIO-Based Methods,** *Ricardo Anido and Ana Cavalli*

Departamento de Ciência da Computação — IMECC
Caixa Postal 6065
Universidade Estadual de Campinas
13081-970 – Campinas – SP
BRASIL

reltec@dcc.unicamp.br