State of the Art and Requirements in GIS*

Nieves R. Brisaboa, José A. Cotelo Lema, Miguel R. Luaces, José R. Viqueira

Laboratorio de Bases de Datos.
Facultade de Informática. Universidade da Coruña.
Campus de Elviña s/n. 15071. A Coruña. España.
brisaboa@udc.es
{jaclema, joserios, luaces}@mail2.udc.es

Abstract. Geographic Information Systems (GIS) provide an appropriated environment for the capture, storage and management of both alphanumeric and geographic characteristics of application objects. These systems have experienced a sensible development during last years, mainly due to two reasons: The performance improvements in conventional computer systems and the arrival in the market of new GIS development tools. In this paper we show the state of the art on this field, where unfortunately, current computer science students receive little training, despite the growing demand of GIS professionals in the market.

1 Introduction

The exponential improvement in the performance of computer systems has made possible the appearance of new tools to manipulate geographic characteristics of objects, allowing their graphical representation on a map. *Geographic Information Systems* (GIS) make an step forward over traditional information systems, offering an appropriated environment for the capture, storage and management of both alphanumeric and geographic information. By geographic information we mean here information about the spatial location of objects. This information can be as simple as the position in the map of all the hospitals on a country or as complex as the partition of the country's land with regard to the kind of vegetation that grows on it.

A GIS must allow the efficient exploitation of all the information it manages, not only providing spatial operations for geographic data, but also allowing to browse and analyze these data graphically. Such graphic aspect simplifies the identification of geographic relationships between objects. Examples of application domains for GIS are, among others, cadastre management, sanitation and communication networks, computer assisted navigation, and decision support systems.

It is important to distinguish clearly between a GIS application and a tool to develop GIS applications (in the following GIS development tool). The latter provides the developer with the capabilities required for capturing, storing and managing both

* Work partially supported by the Comisión Interministerial de Ciencia y Tecnología (CICYT, references TEL99-0335-C04-02 and TEL99-0335-C04-03) and the Xunta de Galicia (PGIDT99PXI10502A).

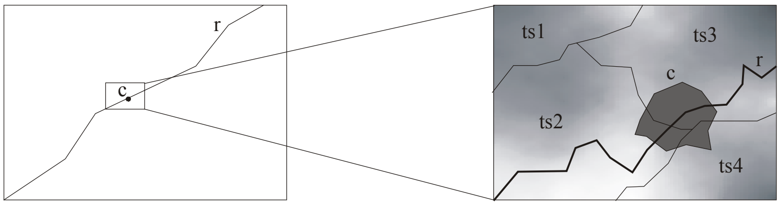
alphanumeric and geographic information. The former makes use of such capabilities to provide the users with a more specific environment adapted to the problem. This difference is similar to the one between traditional DBMS and information systems.

In this paper we report on the *state of the art* in GIS, describing the evolution that GIS applications have experienced over time and analyzing the functionalities that GIS development tools should provide. Current research areas are also explored. The paper is structured as follows: Section 2 introduces basic concepts on geographic data. Section 3 makes an overview of the evolution of GIS development tools and applications. Section 4 shows two different approaches for modeling information in GIS. Section 5 analyzes the basic operations that a GIS should provide, whereas Section 6 shows the interface requirements of GIS applications. Section 7 points out new research lines in GIS and spatial databases and finally Section 8 concludes the paper.

2 Basic Concepts

We call *geographic space* the space of coordinates (usually R^2) where geographic data are represented. This space is, usually, either Cartesian (the GIS uses a flat model of the earth) or geodesic. Relatively often, a GIS application needs to store some alphanumeric value, of a given domain, for each point of such space. We call this kind of data *geographic space attributes* (GSA for short), also known as field-based geographic information. A GSA can be either continuous (if the associated value can change smoothly over the space) or discrete (if the domain of the associated value is discrete). Examples of GSAs are *kind of vegetation* (discrete) and *soil salinity* (continuous). Besides GSAs, a GIS allows the storage of geographic properties of application objects, also known as object-based geographic information. In this context it is convenient to introduce the following concepts:

- *Geographic object or entity*. It is an application object for which the GIS stores both alphanumeric and geographic properties.
- *Geographic attribute*. Represents one geographic property (position, extension, etc.) of an object. A *Geographic domain* is the set of values that a geographic attribute may have. Examples of geographic domains are:
  - *Point*: Represents a single point in the space, for example the location of a farm.
  - *Line*: Represents a sequence of contiguous points in the space (a curve). An example of such value is the course of a road.
  - *Region*: Represents a connected area in the space. A parcel and the area belonging to a municipality are examples of such values.
  - *Collection*: They are sets of the previous. They might be either homogeneous, where all the elements belong to the same type, or heterogeneous, where elements of various types coexist. An example of the former is the area covered by the snow in a country. The hydrography of a given area, including both rives (lines) and lakes (regions) is an example of the latter.
  - *Partition*: Represents an element of one partition of the space in disjoint areas. For example the division of a province into its municipalities.



(a) Scale E1 (b) Scale E2

Fig. 1. Examples of geographic data represented at two different scales.

Figure 1 shows a representation of two geographic objects and two GSAs at two different scales. Thus, at scale E1 the geographic object "c" (a city) is represented by an attribute of type *Point*, whereas the geographic object "r" (a river) is represented by an attribute of type *Line*. When scale E2 is used, the representation of object "c" changes from a value of *Point* type to a value of a different attribute of *Region* type. Object "r" is represented at this scale by a more precise value of the same data type *Line*. In addition, the discrete GSA "kind of vegetation" (splits the space into the vegetation types ts1, ts2, ts3, ts4) and the continuous one "salinity" (dark colors represent lower soil salinity) are also depicted at scale E2.

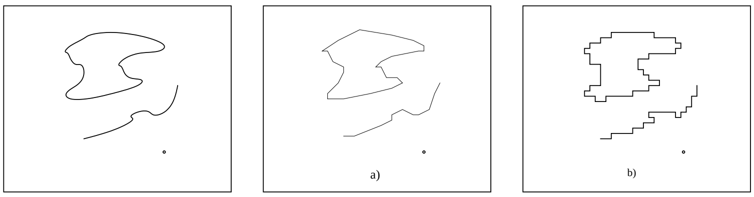


Fig. 2. Representation of spatial values using a) *realms* and b) *quanta*.

Conceptually, the values of the previous domains are non-empty and possibly infinite subsets of geographic space. However, in order to represent such values in a computer system, infinite sets must be replaced by some finite representation. Thus, values conceptually defined over R^2, must now be redefined in a finite grid of points, usually using linear approximations [10]. The way these representations are defined, their semantics and even the number of domains defined, changes from author to author. For example, in [14] spatial values are approximated by finite sets of minimal spatial elements called *quanta*. In [12], spatial values are represented using elements of a *realm*, which is basically a finite and not fixed set of points and line segments. There must not exist a pair of segments in the *realm* that intersect, and for any point *P* and segment *S* of the *realm* such that *P in S*, *P* must be an end point of *S*. Figure 2 shows how geographic values are approximated in both approaches.

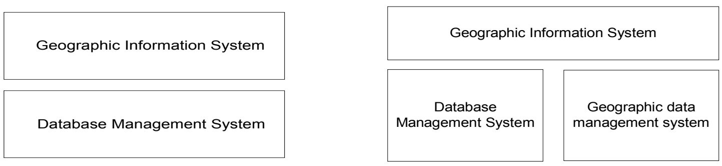
3 Evolution of GIS development tools

According to [4], GIS development tools evolved through three generations:

First generation. In tools of this generation geographic data (and associated indexes) are stored in system files, using proprietary formats. Thus, the GIS tool is the only one capable to interpret and manipulate them. Besides, the GIS tool can only process spatial data. Another application working over it is used to combine spatial and alphanumeric data. Both data model and interface of the GIS tool are proprietary. Finally, the GIS application itself must deal with security, recovery and integrity aspects, given that the GIS development tool does not provide such support.

Second generation. This generation appeared with the intention of integrating geographic and alphanumeric data. Two different architectures are clearly identified:

- *Layered* architecture. It follows the approach of adding spatial support to an existing DBMS. The GIS functionality is implemented as a layer over the DBMS, as shown in Figure 3(a). Geographic data are managed by the DBMS, making possible to use its support for transaction management, security, recovery, etc. In a first approach, geographic values are decomposed into simpler components, represented by conventional values and stored each one in a different tuple. This is clearly expensive in terms of computation. A second approach uses large object data types to store geographic values. It is more efficient, however, geographic data is still opaque to the DBMS, disallowing the optimization of geographic queries and the use of advanced spatial indexes.



a) Layered architecture. b) Dual architecture.

Fig. 3. Second generation architectures.

- *Dual architecture*. It is typically used in GIS systems of cartographic or CAD environments, where alphanumeric information is associated with geographic attributes of the application domain. A DBMS is used for the management of alphanumeric data and a second subsystem (the existing cartographic or CAD system) for the management of geographic data, as shown in Figure 3(b). Both subsystems are integrated through a layer that provides also the graphical user interface. It allows a more appropriated representation of the geographic data as well as the use of more suitable spatial indexes. However, queries become much more complex and their global optimization is impossible. Moreover, the subsystem for the management of geographic information must implement again standard functionality of DBMS as transactions, security, etc.

In general, in second generation GIS development tools the semantic and structure of geographic data are completely unknown for the DBMS, reducing the possibilities of query optimization. Besides, the DBMS must know the behavior of the GIS tool and its interactions with the DBMS in order to make an appropriated database management. Finally, the GIS application defines a specific database scheme to be used, making difficult the integration with pre-existent data.

Third generation. Third generation GIS development tools appeared in the market last years, aiming to provide a full integration of geographic data into the DBMS. They are designed as modules that extend existing DBMSs with new spatial data types, operators and indexes. Given that the DBMS understands natively spatial data, it becomes responsible of all the management tasks (transactions, security, query optimization, etc.). Besides, queries on any kind of data are written with the same spatially extended query language. Fortunately, GIS development tools seem to be moving towards this third generation of tools. For example, the last spatial extensions of Oracle, originally belonging to the second generation (layered architecture), have already done it. This evolution, however, seems to be slower in GIS development tools using a dual architecture, probably because of the special way they see the relationship between spatial and alphanumeric data.

4 Data Modeling

In this section we shall illustrate, with the relational model, the difference between the modeling of data in cartography oriented and spatial databases oriented approaches.

LAYERS:

CITY_E1 (city_cod, municip_cod, name, point)	ROADS (road_cod, line)
CITY_E2 (city_cod, municip_cod, name, region)	SALINITY (salinity, partition)
SOIL_TYPE (soil_type, partition)	

RELATIONS:

MUNICIPALITY (municip_cod, population)

Fig. 4. Layer based representation on cartography oriented GIS development tools.

In cartography oriented GIS development tools, the geographic information is structured in *layers*, defined as relations (in the relational model sense) with just one geographic attribute. A layer represents all the geographic objects of one given type (e.g. roads), one object per tuple. Continuous and discrete geographic space attributes (GSAs) generate also the corresponding *layers*, one per GSA. Figure 4 shows how the example in Figure 1 can be modeled using *layers* and alphanumeric relations. An example of extension to the relational model following this approach is shown in [5].

In GIS development tools oriented to spatial databases no special treatment is given to entities or relations having geographic attributes. With regard to GSAs, they are represented as in the cartography case (one relation per GSA). Figure 5 shows how the example in Figure 1 can be modeled following this approach. It is important to remark here that some geographic relationships between geographic objects (e.g. a road connects two cities) do not need to be explicitly stored in a relation. Instead, they can be retrieved using geographic operators. Relational extensions following this approach are shown in [9, 14, 19].

RELATIONS:

CITY (city_cod, municip_cod, name, point_e1, region_e2)	
SOIL_TYPE (soil_type, partition)	SALINITY (salinity, partition)
ROADS(road_cod, line)	MUNICIPALITY (municip_cod, population)

Fig. 5. Representation on spatial databases oriented GIS development tools.

Extensions to other logical models (other than the relational one), like the nested relational [1], object-relational [20] (used in the spatial extensions of most commercial databases [17, 13, 4]), object-oriented [21], etc., have been proposed in the spatial database literature. The inclusion of spatial data types in the object-oriented model is considered in the OpenGIS standard [16].

5 Data Management

A key aspect in a GIS development tool is the functionality offered for both alphanumeric and geographic information retrieval.

5.1 Required Functionality

The high diversity of GIS application domains prevents the definition of a complete set of tools covering all the required functionality. Instead, the solution is to provide developers with a short set of basic tools used, into a development environment, to construct tools adapted to a specific problem. Such basic functionality includes:

- Checking the topological relationships between geographic attributes [2, 8].
- Computing numerical properties of spatial attributes (e.g. *area*, *length* or *distance*).
- Building new geographic values from existing ones (e.g. *union*, *buffer areas*, etc.).
- Manipulating sets of alphanumeric and geographic data, like the *reclassification* (merges areas with identical alphanumeric characteristics) and *overlay* (combines geographic and alphanumeric data from two different layers) operations [5].
- Complex and specific spatial analysis, as the shortest path in a network of lines.
- Statistical studies of continuous GSAs, like interpolation (building a relation of temperatures from a set of measurements) or discretization (to divide the space, from a layer of temperatures, in areas representing 5 degrees intervals).

In cartography and CAD oriented GIS development tools some specialized algebra must be defined over layers [5]. Considering that a layer has one and only one geographic attribute, a projection can only remove non-geographic attributes. A second operation can be provided if relations (without geographic attributes) are also allowed in the model, to get a relation from a layer by removing the geographic attribute. Operations as *Cartesian product* and *join* cannot be applied between two *layers*, because they would return a relation with two geographic attributes, which is not *layer*. Of course, they can still be applied between a relation and a *layer*.

In GIS development tools based in spatial databases all operations that can be applied to relations having only alphanumeric attributes must also be applicable to relations with spatial attributes. However, such operations must be extended with operations providing geographic functionality. Mainly, they include *spatial predicates*, which can be used in the conditions of *selection* and *join* operations, *spatial functions* and new relational operations (*Reclassification* and *Overlay* are defined as relational operations). New relational operation for spatial management must be applicable also to relations having only alphanumeric attributes. Spatial extensions to the relational algebra were introduced in [9, 14, 19]. Other models, as the object oriented model or the object-relational model, have also been extended with new geographic classes. Finally, [12] describes an algebra for the direct manipulation of spatial objects, which can be included in several of the existing logical models.

5.2 Query Languages

Most commercial GIS development tools provide wide sets of commands, to implement operations between layers. In the case of a relation with several geographic attributes, it can be seen as several layers, one per geographic attribute. Some examples are Arc/Info 7 and Geomedia 3.0. The expressiveness and friendliness of this set of commands are far away from the ones provided by classical query languages like SQL. Several extensions to SQL have been proposed in the spatial database literature [7, 18]. In the relational model, for example, SLQ92 can be extended with new spatial data types, predicates and functions to be used in queries (mainly in the WHERE clause), and new operations, like OVERLAY, to be added to the already existent UNION, JOIN, etc. With regard to object-relational model, the SQL99 standard already provides support for spatial attributes, whereas extensions to the object oriented model should be based on the OQL language and on the OpenGIS standard. The latter defines a set of basic geographic data types that GIS applications and spatial databases should be able to manage, as well as the basic set of operations that should be provided over them.

6 Interfaces

With regard to the interfaces of GIS applications, one must consider two different aspects, namely user interfaces and interfaces with other systems (interoperability).

6.1 User Interfaces

Since the main goal of GIS applications is data analysis and decision-making, they must allow qualitative analysis of geographic data. Whereas in traditional DBMS all the attributes can be easily entered using the keyboard and the results can be shown as text, the data managed by a GIS application must be shown using a graphical interface. As pointed out in [6], qualitative analysis is usually made by visual comparison, which is difficult to do if the information to compare cannot be seen simultaneously. Therefore, the GIS interface must be able to show several maps and alphanumeric tables simultaneously. This can be achieved using *layers*.

The concept of layer here is similar to that of cartography-based GIS tools. Geographic objects are represented either as opaque, semitransparent or as their outline. The layer will be transparent where there are no objects, so that layers below can be seen. To represent a discrete GSA, all points that have the same value are grouped into a geographic object. To represent a continuous GSA, each point of the space is given a color proportional to its value. Colors, styles, patterns, etc. used to depict a graphic object in a layer might be changed by the user at any time.

The user must have a means of quickly accessing the attributes of a geographic object, for example by labeling its graphical representation with selected alphanumeric values. It must also be possible to select and inspect particular graphic objects. Besides, to avoid the need of clicking in the map with pinpoint accuracy, the user interface must allow some kind of rough selection. It should also provide support for performing queries graphically.

It must be possible to use different representations for the same object at different scales. For example, one could use a simplification of the object contour or a new representation of the object depending on the scale. This is known as the generalization problem [22, 23, 24].

6.2 Interoperability

With the grown of the GIS market and the generalization on the access to information through data networks (e.g. Internet), the capability of accessing information stored in several systems and coming from different sources has become a must in the GIS community. Therefore, a GIS development tool should allow the (easy) development of GIS applications capable of simultaneously connecting to several sources of information, located in different computers and using different storage methods. This means not only that the access to remote sources should be as transparent as possible, but also that the layer structure should not be fixed in advance. Moreover, GIS applications must be able to work simultaneously with data represented using different coordinate systems. Given that a significant amount of spatial data are captured using CAD tools, their formats must be easily imported. Finally, the capability of making public the information managed by the GIS application should not be neglected. This includes the capability to export data under the more important standards (e.g. OpenGIS [16]) and the capability of allowing the public to perform queries to the GIS (e.g. using the WEB).

7 New Research Lines

Among the promising areas for research in GIS we would like to stress the followings:

- Indexing. The still growing requirements in the amount of spatial data that a GIS must manage demands further improvement of the existing spatial indexing technology.
- Data models and ontologies. The use of finite representations for spatial data generates numerous robustness and consistency problems. Although several solutions have been proposed for reducing such problems [3, 12, 14, 15], there is still a lot of research to be done in that direction.
- Time support in GIS and spatio-temporal DBMS. An interesting aspect that challenges the GIS community is to provide support for time dimension. This includes the representation, management, and querying of spatio-temporal information for both information that changes at discrete time instants (e.g. the area of a parcel) and information that changes continuously [11] (moving objects).
- Interoperability. The integration of geographic information provided by different sources, represented at different resolutions or modeled using different philosophies (e.g. whether one given class of objects is represented as a region or as a point) need further research to improve the interoperability between GIS systems.
- Indexing and *joins* of heterogeneous attributes (e.g. when the involved spatial attributes are represented using different types of geographic attributes).
- Manipulation of geographic space attributes and networks. Most of research in GIS has focused in object-based geographic information. However, there is still research to be done in the management of geographic space attributes (GSAs) and in the development of GIS tools specialized in network management.
- Integration of fuzzy information management in GIS.

8 Conclusions

In the following years it is expected a drastic increment in the application of GIS solutions to several application domains due to their capability of easily showing the spatial relations between geographic objects through their graphical representation on a map. Moreover, the evolution experimented by GIS development tools makes easier and faster the development of effective and suitably adapted GIS applications. The full migration of GIS development tools towards their 3rd generation, together with new advances in the research areas described in Section 7, should provide even bigger functionalities to the GIS area, mainly in certain application domains as cadastre, network management, etc., making the GIS field even more promising.

For all these reasons we expect GIS to be a field with a high demand of professionals in the following years and regret the (in general) lack of training of current computer science students on that field.

References

1. Chan, E. P. F., Zhu, R.: QL/G – A Query Language for Geometric Data Bases. First Int. Conference on GIS, Urban Regional and Environmental Planning. (1996) 271-286.
2. Clementini, E., et al: A Small Set of Formal Topological Relationships Suitable for End-User Interaction. SSD'93. LNCS 692. Springer-Verlag, (1993), 277-295.
3. Cotelo Lema, J. A., Güting, R. H.: Dual Grid: A New Approach for Robust Spatial Algebra Implementation. FernUniversität Hagen, Informatik-Report 268, (2000). To be published by the Geoinformatica Journal.
4. Davis, J. R.: IBM's DB2 Spatial Extender: Managing Geo-Spatial Information within the DBMS. http://www.ibm.com/software/data/pubs/papers/. Tech Report IBM Corp. (1998).
5. Delis, V., Hadzilacos, Th., Tryfona, N.: An Introduction to Layer Algebra. Proc. 6th International Symposium on Spatial Data Handling (SDH'94). (1994) 1020-1041.
6. Egenhofer, M. J.: Interaction with Geographic Information Systems via Spatial Queries. Journal of Visual Languages and Computing 1, Academic Press Limited (1990) 389-413.
7. Egenhofer, M. J.: Spatial SQL: A Query and Presentation Language. Transactions on Knowledge and Data Engineering, Vol. 6(1) (1994) 86-95.
8. Egenhofer, M. J., Herring, J. R.: Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases. Technical Report, Department of Surveying Engineering, University of Maine (1990).
9. Gargano, M., Nardelli, E., Talamo, M.: Abstract data types for the logical modeling of complex data. Information Systems 16(6) (1991) 565-583.
10. Güting, R. H.: Special Issue on Spatial Database Systems: An Introduction to Spatial Database Systems. VLDB Journal: Very Large Data Bases, 3(4) (1994) 357-399.
11. Güting, R.H., et al.: A Foundation for Representing and Querying Moving Objects. ACM Transactions on Database Systems 25:1 (2000) 1-42.
12. Güting, R. H., Schneider, M.: Realm-Based Spatial Data Types: The ROSE Algebra. VLDB Journal 4, (1995) 100-143.
13. Illustra 2D Spatial Datablade (Release 1.3) Guide. (1994).
14. Lorentzos, N. A., Tryfona, N., Rios Viqueira, J. R.: Relational Algebra for Spatial Data Management. ISD'99. LNCS 1737, (1999) 192-208.
15. Muller, V., et al: Virtual Realms: An Efficient Implementation Strategy for Finite Resolution Spatial Data Types. SDH'97. Taylor & Francis, (1997) 697-710.
16. OpenGIS® Abstract Specification (V 4). (1999). http://www.opengis.org/techno/specs.htm.
17. Oracle 8i Spatial User's Guide and Reference. Release 8.1.5. (1999).
18. Roussopoulos, N., Kelley, S.: PSQL: An Inexpensive GIS Solution for Oracle. Advanced Communications Technology Inc. Technical Report, (1995).
19. Scholl, M., Voisard, A.: Thematic Map Modeling. Buchmann, A., et al. (Eds.): Design and Implementation of Large Spatial Databases. LNCS 409. Springer-Verlag (1989) 167-190.
20. Vijlbrief, T., van Oosterom, P.: The GEO++ system: An Extensible GIS. Proc. 5th Int. Symp. on Spatial Data Handling (SDH'92), Charleston, SC (1992) 40-50.
21. Voigtmann, A., et al: An Object-Oriented Data Model and Query Language for Geographic Information Systems. Tech. Report 15/95-I, Univ. of Munich (1995).
22. Voisard, A.: Towards a Toolbox for Geographic User Interfaces. SSD'91. (1991) 75-97.
23. Voisard, A.: Designing and Integrating the User Interface of Geographic Database Applications. ACM Int. Conference on Advanced Visual Interfaces (AVI) (1994) 133-143.
24. Voisard, A.: Mapgets: A Tool for Visualizing and Querying Geographic Information. Journal of Visual Languages and Computing, 6(4), Academic Press, (1995) 367-384.