CONCEPTUAL MODELING OF GEOGRAPHIC APPLICATIONS

Silvia Gordillo
LIFIA—University of La Plata, Argentina

Robert Laurini
Claude Bernard University of Lyon—INSA, France

Keywords: geographic information systems, databases, conceptual modeling, object-oriented modeling

Contents

Summary

It is widely known that modeling and design are key activities in developing information systems. A good design model helps to achieve modularity and ease of evolution and maintenance. Unfortunately these aspects have often been neglected in the area of geographic information systems (GIS), as performance and accuracy issues have been favored instead of more conceptual ones.

In this paper a survey of existing approaches for modeling and designing GIS applications is presented. The nature of design activities in GIS applications is first discussed; some notations and methods related with GIS design are then analyzed; finally an object-oriented approach, based on design patterns and integrating discrete and continuous data into an uniform design model is shown.

1. Introduction

The development of a GIS application involves four relevant activities: data input, data management, data manipulation and analysis, and data output. The major bottleneck is generally found in the first step, since the acquisition of accurate data is critical for the operation; since this process is the most expensive in the whole development, design aspects are usually ignored. However, designing GIS is intrinsically difficult because of the characteristics of spatial information and the varied functionality that users need from these systems. They not only have to deal with complex domain and mathematical models (such as for example floods and their impact on cities and fields) as in conventional information systems; they also have to take into account the spatial nature of the information involved in these systems (see *Spatio-Temporal Information Systems*). Representing spatial information usually involves the acquisition, storage, and manipulation of geographical locations of real world entities. These locations cannot be manipulated as simple attributes; they subsume a set of aspects that are needed to obtain the expected application's capabilities. For example, it is necessary to know the reference system in which locations should be interpreted; the treatment of positional attributes expressed in latitude/longitude is different from those defined in the Cartesian system.

Locations should also be expressed depending on the individual aspects of the entity they represent: positions of entities whose geometry differ are usually defined in a different way. Depending on the application, entities might be static (i.e. their location does not change with time), or changes in their locations might be infrequent, or their location might change frequently or continuously, in which case it is necessary to manipulate their evolution (and therefore handle additional concerns, such as for example time). Representing locations also involves defining operations involving spatial objects: distances, area overlay, the union and intersection of areas, and so on.

Finally positions, and as a consequence spatial objects, are related, and they are also related to other application entities (spatial or not). These relationships should clearly reflect the domain model and should be designed carefully.

This work presents a discussion about the importance of conceptual modeling approaches in the GIS domain. Their use helps to produce clear and consistent designs in GIS applications. Section 2 discusses the different elements that have to be taken into account when working with geographical entities. Section 3 describes some of the mainstream modeling approaches targeted to the GIS domain. Section 4 presents a general framework for modeling and developing geographic applications. Finally, in Section 6 some concluding remarks are presented.

2. Modeling GIS Applications: The Nature of Spatial Information

As with entities in most information systems, entities in GIS applications should model entities in the real world; thus there is a set of relevant information related with these entities that should be represented in GIS software. This information consists of nonspatial and geographical data. For example, in a cadastral application, data about property owners (name, social security number, etc.) will be modeled, and when geographic information is considered, some additional data must be taken into account (the location of properties, for example) since it strongly defines the functionality of the system.

As said before, the representation of objects' locations is crucial; however this task is not so easy. Locations are not merely numbers; they abstract complex objects, and usually additional information is needed to interpret and manipulate them. For example, there is a difference between manipulating objects in a Cartesian system (defined by two or three coordinates) and dealing with them in a geodetic system (which expresses coordinates as latitude/longitude/elevation). It is necessary to specify the reference system in which these numbers are expressed. A good modeling approach for dealing with locations is important in order to define application entities completely.

There are two well-known approaches to dealing with spatial information: the raster and the vector models. In the raster model, the area of study is divided into regularly spaced cells; in each cell a code is assigned, indicating the value of the attribute. In the vector model, individual objects are classified according to their "shape"; this means that they are represented in terms of their "geometry", for example as points (zero-dimensional objects), lines (one-dimensional objects) or polygons (two-dimensional objects). There are many data models targeted to describing objects depending on their geometry, such as the topological model, the spaghetti model, and TINs.

Modeling locations also involves defining a set of operations for the manipulation of objects, such as calculating distances among objects. These operations depend on the geometry of objects and on the reference system in which the location has been defined. For example there is a difference in computing a distance between a point and a line or between a line and a polygon; there is also a difference if points are defined in a Cartesian system or in the Geodetic system. These operations are called topological operations and they are an essential part of a GIS application (see *Cartographic Generalization: Interface Issues* for more details on these structures).

To integrate more realism into this model, let us say that the geographical world is not only represented by individual and discrete entities; many times a more comprehensive vision is needed. For example, in order to deal with climatic phenomena such as temperature, pressure, and rainfall the notion of continuity has to be considered, since it is necessary to know the value of a phenomenon at any location of the (continuous) region under study. To represent continuous information, it is usually necessary to perform a discretization process that leads to the representation of objects in term of points, lines and polygons (with further considerations to manipulate continuity).

The discretization process involves storing only a set of values that result from sampling the phenomenon at different geographic locations (the sampled points) and using an estimation method for interpolation, calculating the values at those locations where the real value is unknown. To solve this problem, basically three main aspects need to be considered: the region under study, the estimation method, and the specification of the sample. (For additional details on conceptual modeling of applications, see *Detail Filtering in Geographic Information Visualization*.)

The next section describes different geographic data design models and the way in which they represent each of these characteristics. As will be seen, several classes of models are centered on static, discrete aspects of the world, whereas the trend is to also encompass dynamic, continuous aspects.

3. Conceptual Modeling Approaches

In this section some of the existing modeling approaches used to design geographic applications are described; the goal is to show how these approaches deal with the kind of data described in Section 2. Each approach is presented through the analysis of a specific proposal, which can be considered representative of the approach.

3.1. Conceptual Modeling: The Entity-Relationship Approach

Conceptual modeling in database design often makes use of a formal approach known as entity-relationship modeling. First presented comprehensively by Chen in 1976, but based on some older ideas, it is a means to organize and schematize information. We first present the formalism and then give an example that will be used often in this paper. In this formalism the basic components are:

- entities
- classes of entities
- relationships between entities or classes of entities
- attributes for both entities and relationships
- cardinalities of relationships
- integrity constraints

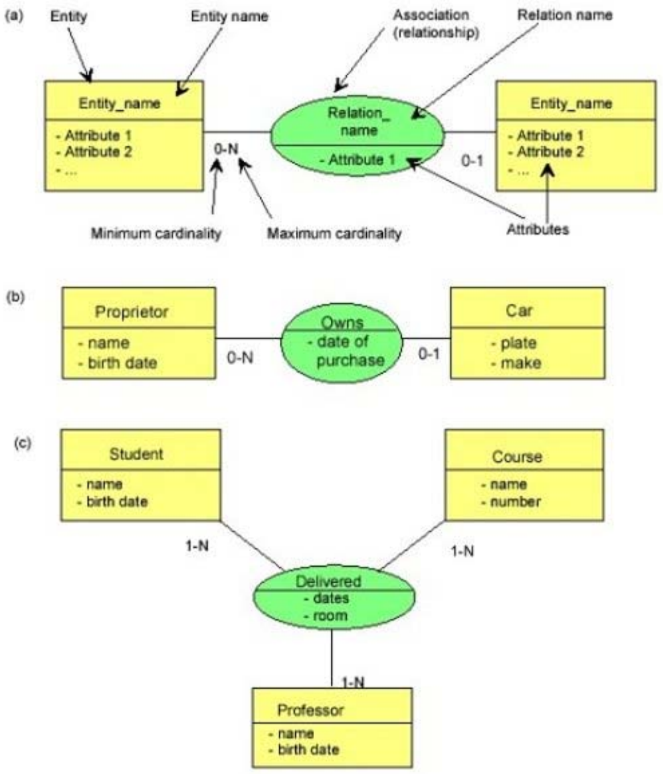

Figure 1. Examples of entity-relationship diagrams. (a) Nomenclature for entity-relationship diagrams. (b) Example of a binary relationship. (c) Example of a ternary relationship.

An entity is a person, place, thing, or event. Mr. Green's parcel, the building 2314, the car numbered 34 HP89, the flight 123, are entities in the real world. But as first illustrated in Figure 1, entities can be regrouped into classes.

Relationships, definable for single entities or classes, are the associations between phenomena: for example the land parcel contains a house, the car has an owner, and the flight goes between Greenland and Fiji. Both entities and relationships can hold attributes, their special characteristics, such as the size of the parcel, the owner of the car, or the flight duration.

Generally, an association is binary, that is, linking only two entities, but there may be more involved. The degree of relationship expresses this number of linkages. Cardinalities are expressed by four numbers defining the minimum and maximum number of entities occurring in a relationship, in both the forward and reverse senses. Usually the minima are 0 or 1; when the maximum is unknown, the letter *N* is used. In some special cases, one may need to use a summary statistic, for example the average number of vehicles. An integrity constraint can be defined as a predicate (value or symbol) that must be matched in order to confer integrity on the model. Integrity constraints can be applied to attribute values or to attribute definitions, but the most important are cardinality constraints.

Consider a simple example dealing with proprietors and cars. A proprietor can own several cars (minimum 0 and maximum *N*), and cars can be owned by zero or one or more (joint owners) people. So the entity-relationship diagram can be designed as in Figures 1b and 1c. As attributes in this example, one can mention the owner's name, his or her date of birth, and so on; the make of the car, or its license plate number. One can also consider attributes in the relationship such as purchasing dates.

In the visual convention used here (Figure 1a), one of several in customary use, an entity or class of entities is depicted by a rectangle and a relationship by a circle or an ellipse linking two or more entities or entity classes. Attributes are shown as lists in the rectangle boxes, or attached to the relationship ellipses. Figure 1c illustrates a ternary relationship among three entity classes of student, course, and professor. More generally, it is occasionally necessary to deal with *n*-ary relationships, where *n* denotes the number of classes.

A more comprehensive example, Figure 2, relates to a city containing land parcels as a principal spatial phenomenon. A description of parcels and streets (referred to here and later as our toy cadastre) and of landowners is prepared in entity-relationship form (Figure 3). A parcel is limited to the range from at least three up to *N* boundary segments; a street is limited to a minimum of two to an unknown upper limit of *N* segments. A segment can delimit one or two parcels and can border one street or none at all. Moreover, a segment is composed of two and only two end-points (vertices), and an end-point can be connected with one to many segments. Here, we can illustrate a cardinality constraint about segment cardinalities. That is, a segment can be bordered by two parcels or by one parcel and one street. In other words, the sum of cardinalities must always be equal to two.

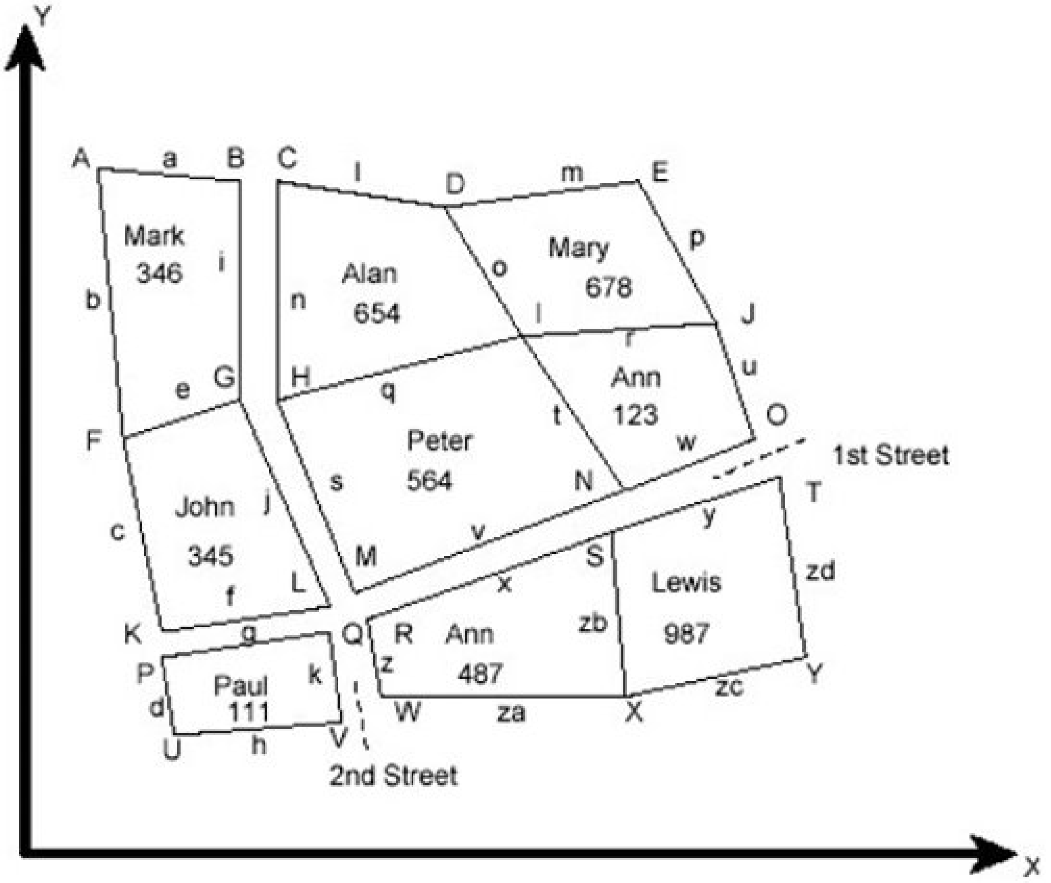

Figure 2. An example of land parcels: toy cadastre.

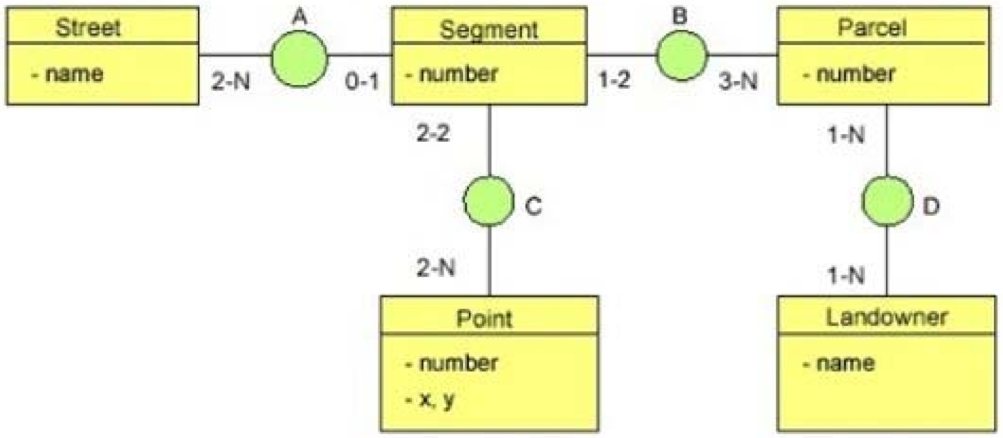

Figure 3. The entity-relationship diagram for land parcels. A: streets have edges (segments); B: parcels have boundaries (segments); C: line segments have two endpoints; D: parcels have owners, and people own land.

The entity-relation modeling design approach (and its variations) is a general tool in the business world for designing databases. Only a handful of examples of its use for spatial information systems have been published, suggesting limited use in spatial information systems design. (See *Spatial Query Languages* for one such example.) However, it would appear that the inherent complexity in spatial problems should lead to more use of design tools like these semantic data models, if only to come to grips with that complexity. Formalized data modeling can provide a basis for discussion and then refinement; and it is a structured foundation for designing a database by implementation techniques such as those discussed below. It is most valuable for complex situations, if only to reveal how complicated reality and user perceptions are.

3.2. Models Based on Entity-Relationship Extensions

A common kind of modeling approach is based on extending the entity-relationship (E-R) model with specific constructs, to encompass spatial characteristics, in particular geographic attributes and relationships. This kind of approach is based on defining a conceptual model of the world in terms of entities (described by a set of attributes) and relationships among these entities. This leads to a rich model that can easily be implemented using a relational database. Since geographic applications always hold the specification of a set of spatial characteristics, like locations, description of geometry, topological properties, some extensions to the basic E-R model may be useful to model those features.

An interesting example is the Giser data model that extends the enhanced entity-relationship model (EER) to specify continuous fields. The EER model extends the basic E-R model with the concept of superclass/subclass and the related concepts of generalization/specialization and category; it also includes the mechanism of inheritance.

The Giser model approach explicitly represents the discretization process used to unify the continuous and discrete visions of the world. Giser is based on four concepts: Space/Time, Features, Coverages, and Spatial Objects. “Features” represent geographic phenomena as continuous fields varying in space and time. A Feature instance might have a unique name (“proper Feature”) or might be identified by its location in space and time (“common Feature”).

“Coverage” instances are the result of the discretization process of a Feature. They consist of a set of “Spatial Object” instances. Spatial Object instances, in turn, have an attribute representing a geometry and/or a topology. Finally, “Space/Time” objects specify the space and time a Feature or a Spatial Object occupies.

The extended E-R model constitutes a good solution since it is simple to use and can easily be mapped to a commercial relational database. However, only static aspects are reflected since the model does not provide tools to represent dynamic characteristics such as the behavior of entities.

-
-
-

Bibliography

Aronoff S. (1988). *Geographic Information Systems: A Management Perspective*, 294 pp. Ottawa, Canada: WDL Publications. [This book provides an introduction to the main concepts of geographic information systems.]

Caron C., Bedard Y., and Gagnon P. (1993). MODUL-R: Un formalisme individuel adapté pour les SIRS. *Revue de Géomatique,* **3**(3), 83–306. [This work presents an extension of the entity-relationship model by proposing pictograms, applied only to entities. It distinguishes between spatial and temporal pictograms.]

Chen P. (1976) The entity-relationship model: towards a unified view of data. *ACM Transactions on Database Systems*, **1**(1), 9–35. [This is considered to be the seminal paper for the description of the entity-relationship model].

Gamma E., Helm R., Johnson R., and Vlissides J. (1995). *Design Patterns: Elements of Reusable Object Oriented Software*, 395 pp. New York: Addison Wesley Professional Computing Series. [This is the main reference for design patterns in the software field.]

Kosters G., Pagel B., and Six H. (1995). Object-oriented requirements engineering for GIS applications. *Proceedings of the ACM-GIS95: Third International Workshop on Advances in Geographic Information Systems* (ed. R. Laurini, P. Bergougnoux, K. Makki and N. Pissinou), November 1995, pp. 61–68. [This work presents an extension to the object-oriented analysis method to define whole-part relationships and graphs.]

Laurini R. and Gordillo S. (2000). Field orientation for continuous spatio-temporal phenomena. *International Workshop on Emerging Technologies for Geo-Based Applications*, May 22–25 2000, Ascona, Switzerland, pp. 77–101. [Describes an object-oriented architecture to model and design geographic applications and in particular continuous fields.]

Laurini R. and Thompson D. (1992). Fundamentals of Spatial Information Systems, 680 pp. London, UK: Academic Press. [A classic in the design of geographic information systems.]

Lbath A., Aufaure-Portier M.A., and Laurini R. (1997). Using visual language for the design and query in GIS customization. *Proceedings of the 2nd International Conference on Visual Information Systems*, San Diego, December 1997, pp. 197–204. [Analyses the use of pictograms as a way to visualize spatial information. It provides extensions for using pictograms at attribute level. It is the basis of the framework OMEGA described in Section 3.]

Lopes De Oliveira J., Pires F., and Bauzer Medeiros C. (1997). An environment for modeling and design of geographic applications. *GeoInformatica*, **1**(1) (April), 29–58. [Deals with the use of design environments to help developers in the construction of spatial databases. It introduces UAPE and its main features. UAPE was chosen as a typical example of geographical data modeling computing environments.]

Shekhar S., Coyle M., Goyal B., Liu D., and Sarkar S. (1997). Data models in geographic information systems. *Communications of the ACM*, **40**(4) (April), 103–111. [Defines GISER, a conceptual meta-

model based on the EER to design geographic applications. GISER was chosen to exemplify entity-relationship based modeling approaches.]

Biographical Sketches

Silvia Gordillo received a Ph.D. from INSA-Lyon, France. She has an M.Sc. in Software Engineering from La Plata University, Argentina. She is head of the Geographic Information Systems project at the Laboratory for Research and Training in Advanced Information Systems (LIFIA) of La Plata University, and Professor at the School of Computer Science of the University of La Plata,. Her main areas of interest are databases and geographic information systems. She has been a member of the program committee of many conferences including the ACM Workshop on Geographic Information Systems, and the Latin-American Conference on Computer Science.

Dr. Robert Laurini is the head of the Laboratory for Information System Engineering (LISI), shared by the National Institute for Applied Sciences (INSA) and Claude Bernard University of Lyon, France, and head of the SIGMA/CASSINI, French CNRS Research Group on Geographic Information Systems. He is a Professor at the Computing Department of the Institute of Technology, Claude Bernard University of Lyon. Previously, he carried out research at the University of Cambridge, Martin Center for Architectural and Urban Systems, UK (1976–1977) and at the University of Maryland (Center for Automation Research, USA) (1986–1987). He is a member of the ACM and of IEEE Computer Society. In November 1998, he received an ACM award for Technical Contributions and Leadership. He is Vice-President of the UDMS (Urban Data Management Society), the European association whose goal is the promotion of urban information systems, and also Vice-President of SORSA (the Spatially-Oriented Referencing System Association). He is the associate editor of the *Journal of Visual Computing and Languages* and *Computers, Environment and Urban Systems*. He is a member of scientific committees of the following journals: *GeoInformatica, Sistemi Urbani* (Urban Systems), *URISA Journal* (Urban and Regional Information Systems Association), *Revista Italiana d'Informatica, Revue Internationale de Géomatique* and *Journal of Photogrammetry and Remote Sensing*. He has co-authored five books, including *Ingénierie des Connaissances Spatiales, Fundamentals of Spatial Information Systems* and *Les bases de données en géomatique*. His book *Information Systems for Urban Planning: A Hypermedia Cooperative Approach* was published in February 2001. He also teaches at the University of Venice (IUAV), Italy. He has recently organized or chaired many conferences in the area of GIS and visual languages.