Database Models for Geographic Information System Applications.

Gabriel P. Emerson
University of Minnesota, Twin Cities.
emers089@umn.edu

Abstract
Geographic Information Systems are increasingly widespread in modern applications and research. The field has expanded tremendously since its inception in the 1970s, and has recently gained popularity and practicality with the rise of faster and cheaper processing and display technologies. Today GIS models are used to represent such complex datasets as city planning and public works infrastructure, automated traffic systems and highway management, regional and nationwide emergency response areas, detailed scientific research data, and large scale crop, mineral, and other natural resource surveys. The data storage and processing requirements for GIS systems are quite different from most databases. GIS relys on a high level of spatial and temporal relationship mapping as well as the ability to handle continuous data inputs and queries which do not always decompose well to discrete storage entities. This paper aims to examine the various models which have been proposed and developed to represent dynamic geographic and spatio-temporal data. It is intended as an overview and evaluation of existing research for those who desire a concise survey of models used in GIS.

1. Introduction.

GIS databases can take many forms, largely depending on the types of data being input and stored, and the applications which will be using the data. For simple map layer output to a human user (perhaps as part of a navigation system or aerial photography viewer), the main concern is to choose between raster or vector map displays. Vector maps are more suited to computer-generated or animated displays, where accuracy at different scaling factors is a concern. Raster images are more suited to detailed real-life textures such as aerial photos.

Further complexity comes from the different categories of GIS data models. "Field-Based" models represent areas as continuous surfaces with variable features layered over them. "Object-Based" models treat features as objects placed upon a surface, and independent of their locations. Object-based GIS models have traditionally been more popular, but are somewhat limited in their representations. These models require discretization and interpolation to map continuous surfaces, layers, and features into discrete approximations that can be handled by the processing system. More complex dynamic models must represent time-dependent or event-dependent data such as traffic patterns, weather, and accident reports. For the real-time user in a vehicle or at a traffic control center, this data must be stored, processed, and displayed immediately. However, the long-term storage and statistical processing requirements for such data may be completely different.

Many of these articles deal with methods of extending and expanding layer-based map models to object-based data models. It is interesting to note that neither paradigm has completely overrun the GIS field, and there is yet to be a generally accepted solution to the problem of modeling dynamic spatiotemporal data (4). There still exist a great many production GIS systems which handle exclusively layer-based raster and vector data. These are seen mainly in consumer and web applications where back-end processing and front-end query functionality is limited. The more powerful object-oriented relational models are most often found in industry-specific GIS systems such as public works or natural resource databases.

2. Survey of GIS data model research.

The GODOT model developed by Gunther and Riekert (2) is considered by some to be a major milestone in the GIS field (5). Standing for Geographic Data Management with Object-Oriented Techniques, this model layers extensible spatial representation methods onto a standard object-oriented database. The authors found that most existing GIS systems used proprietary or custom data storage back-ends which were not conducive to inter-interoperability, expandability, or translation into other data formats. By developing GODOT, they hoped to produce a system which could handle complex geo-spatial data in a more useful and flexible format (2).

GODOT is based on a four-layer architecture, with the lowest level being a commercial object-oriented database system (ObjectStore). Above the DBMS sits an extensible kernel and the representation methods for managing spatial and temporal data, as well as standard data objects. The third level consists of query processing and administration components, and the fourth provides C++, UNIX shell, and graphical user interfaces.

The kernel implements the GODOT data model devised by the authors, and defines methods and classes for representing and managing GIS data. The data model offers three categories for objects, including Thematic (representing real-world objects), Geometric (which describe geometric features of objects), and Graphic (used to display thematic objects). A city, for example, is represented by a thematic geo-object connected to graphic objects with represent the geographic location, and geometric objects to represent the shape. By defining each class of object at a high level of abstraction, the authors show how these simple types can be expanded and connected to represent a wide range of real-life geo-spatial objects, abstract concepts like management entities, or even other data such as animal species.

GODOT is the basis of much modern GIS research, and provided one of the first commercial viable object-oriented implementations of a GIS data model. Gunther and Reikert's paper does a thorough job of developing and illustrating the components and operation of the system.

Tang, Adams, and Usery use a similar approach to a feature-based object-oriented database model in their 1996 paper (6). Their work aimed at replacing previous layer-based

GIS data models with a model which could handle semantic relations and interrelations between feature objects as well as geographic relations. While a traditional layer-based model of the time usually depicted a map or set of maps in different layers, the authors' feature-based model could reflect individual geographical features and sets of related features.

In their object-oriented approach, Tang et al considers features as both real-world objects and their computer representations. An encoded digital representation of a feature is called a feature object, and contains relations recording the location and other attributes of the feature. Features can be aggregated into collections of features (such as cities) by moving up in layers of geographic abstraction. The components of a feature-based object (or the columns in a feature relation) are Identifier (assigned by the system), Positional data, Non-spatial data, topological relations, non-topological relations, and methods (such as specifications for computation on or display of the object). The positional data consists of coordinates and elevation, and the non-spatial attributes can be names or values such as population. Topological relations are relations among objects, including boundaries and neighbors. Non-topological relations account for relations such as "above", "below", "is", "part of", etc.

The object-oriented model provides standard features such as inheritance and data encapsulation through classes. Geographic features which are a subset of a class can inherit properties and attributes from their parent class. Such representations are much more flexible and represent real-world data more accurately than flat map files. The authors developed this model during the period when GIS systems were moving from map/layer-only capabilities to more expressive feature and object-based data models, and their paper is just one of several related approaches to the idea of implementing object-oriented data models for GIS platforms. It gives an overview of the advantages of object oriented vs layer-based models, but does not deal with the error checking and interpolation issues that arise when continuous geographic data is converted into discrete objects in this model.

Michael Worboys is one of the leading researchers in GIS database systems, having provided much of the early research into object-oriented structures and functions of extensible GIS (5). In his 2004 paper with Kathleen Hornsby, he outlines a expansion of object-based models which can handles dynamic temporal objects and events. The Geospatial Event Model, or "GEM" is designed to implement events into the object-based approach, and to define event and object relationships (7).

Temporal objects can be transitory in nature, such as street maintenance or route detours, and can be independently occurring or can affect and/or depend on other temporal and spatial objects. In addition, spatial and geographic features and objects which are traditionally thought of as static can in fact change over time. This is seen in the change of a river's floodplain or the spread of urbanization through former farmland. Worboys defines two types of entities to represent such changes; "continuants" (objects which reside at a location for some period of time, such as buildings), and "occurents" (entities which take place over a period of time, such as a construction project). These concepts are represented in the system as "objects" and "events", which are distinct while sometimes having overlapping attributes. Objects and events exist in a "setting" which is either/or a combination of each entity's spatial and temporal coordinates.

Spatial and Temporal settings can be related to other settings of the same type (a purely temporal setting can not directly relate to a purely spatial setting). Spatial settings can have set/subset or location-based relationships, and Temporal settings can have overlapping period relationships as well as interval relationships, the details of which are defined by others. SpatioTemporal settings are considered to be functions mapping temporal to spatial settings, and have a variety of relationships depending on whether the temporal, spatial, or both domains are disjoint.

The heart of this research is the Geospatial Event Model, which defines geospatial event and object instances and components. The model provides a number of functions to compose and relate geospatial objects and events. The model is implemented through an extension of UML which allows users to diagram and define classes and aspects of a GIS system using GEM. The model allows querying of dynamic systems, such as events related to objects, dependent events and objects for other events, and spatio-temporal settings for events. Examples could include "Find the events that are necessary for a vehicle to reach a given destination", or "Find the objects which could prevent a vehicle from traversing a certain region"

Worboys and Hornsby develop a very well-defined model in this paper, covering the most important aspects of the GEM system and its structural and functional composition. While the details are somewhat difficult to follow in parts, the text effectively presents the way in which objects and events differ and interact, and the ways that an object/event model such as GEM can offer improvements over object-only models.

In their paper "Data Models in Geographic Information Systems", Shekhar, Coyle, Goyal, Liu, and Sarkar propose a model that combines the Object and Field-based concepts by explicitly representing the discretization required for GIS data (5). The authors' Geographic Information System Entity Relational model, or "GISER", is designed to handle all stages of a GIS process, from data collection through processing and interpolation to display.

The GISER model is based on the concepts of Space/Time, Features, Coverages, and Spatial Objects. The first concept represents a continuous field in which events can occur, and must be discretized into calendars or geometric representations to be used. Features are relatively static geographic features and phenomena such as rivers, cities, and roads. However, some of these entities (such as crops, bodies of water, or construction projects), are in fact continuous fields, and must be discretized into "coverages" consisting of sets of spatial objects. Depending on the method of discretization and interpolation, the same feature set could derive multiple coverages, leading to the potential for error or multiple interpretations of the data. The authors assume, but do not describe, a "perfect" interpolation model for each coverage which would produce identical results for each derivation.

GISER classified GIS queries into a number of different set operations. "Spatial Selection" returns a subset of data matching a spatial query, such as "Find all gas stations no more than 10 miles from exit 24". "Spatial Join" returns a set of spatial object pairs from two layers, such a query could be "For each interstate, find all Wal-Marts within 1 mile". A "Transformation" query creates a new layer from existing layers of spatial objects, such as rendering a set of vector layers into a downloadable raster map. "Network Analysis" integrates many different data subsets and aggregates data to carry out route-planning, alternate-path computation, and other complex evaluations.

Shekhar et al state that GIS queries based on spatial relationships can be described as either Topological, Directional, and Metric relationships. The first includes connected, disjoint, inside, and other relationships which do not change with scaling or other geometric transformations. Directional relationships indicate the placement of entity pairs or entities from a known reference point, and can include terms like “north_of”, “west_of”, above, etc.. Metric relationships are numerical in nature, such as the distance between two points or features.

This paper accurately describes the requirements and constraints of a GIS model which can handle both Object and Field based data. It also provides a short description of useful spatial dataset queries. It is noticably lacking in that the authors do not describe their interpolation and discretization methods in much detail. The existence of such methods provides the core of their thesis that continuous datasets can be efficiently mapped to discrete data and provide reliable results, but the details are not given.

Ashley Morris outlines a framework for modeling uncertainty in spatial databases in his 2003 paper (3). He lays out the need for representing data with uncertain or imprecise boundaries, such as urban zones or natural features like forests. The definition of “urban” land can be zoning-dependent or building-density dependent, just as forests and other features can be defined by object (tree) density or artificial arbitrary borders. The ability to deal with such areas and features in different ways is important when one wishes to include or exclude outliers from a sample, or to make comparisons on similar features with the same criteria. Features can be represented with or without boundaries and cores, and the boundaries can be defined “fuzzily” through use of different cutoff functions and thresholds.

This paper is both a research survey and a collection of the author’s previous work in this area of study, comparing advances and models proposed by other researchers to the models earlier developed by Morris. The author briefly outlines existing research into storing fuzzy spatial data, but goes on to propose a model which considers representation and querying on such data as well.

Morris lays out two separate categories of fuzzy data, imprecision and uncertainty. Data in each category is separate from the other, and underlying features are preserved through aspects of symmetry, transitivity, and reflexivity defined by well-established functions. By developing the Fuzzy Object Oriented Spatial Boundary and Layer (FOOSBALL) system, Morris claims to provide accurate and reliable results from uncertain and imprecise datasets and queries. FOOSBALL allows the calculation of “degrees of inclusion” for members of a set or feature, letting the user specify exactly what they wish to include in their results. Queries such as “Find all drivable roads in Denali park” can be refined to include “open public roads with some length inside park boundaries, with terrain suitable for a 2WD vehicle”. A large part of this paper details the mathematical basis for the inclusion calculations, based on for spatial and temporal relationships and threshold/relevance values.

Morris asserts that performance has been increased by composing the system solely with vector data, and by assigning dynamic membership values to spatial objects with different alpha-cuts (or cutoff thresholds) rather than assigning membership values to each pixel in a raster dataset. FOOSBALL allows both user and machine-driven alpha-cut definitions for each entity. The author then goes on to illustrate various “fuzzy” query results and compare them to the results obtained with strict relationship boundaries.

This paper provides a compelling argument for the ability to represent and query imprecise datasets. The concept does seem to be well-suited to the domain of spatial and temporal data, especially with certain types of imprecisely-formatted but frequently-executed queries that arise in GIS applications. The demonstrated framework does show advantages over traditional rigid-relationship GIS datasets and queries, and the ability to include user-defined cutoffs is certainly a useful feature.

An application of GIS databases which is becoming more widespread is the incorporation into mobile mapping services and platforms, such as on-board vehicle navigation and handheld GPS devices. Choy, Kwan, and Leong investigate the need for fast-response real-time processing of GIS data for Advanced Traveler Information Systems (ATIS) (1). These systems are typically expected to provide real-time traffic and routing data for travelers, as well as displaying query results for roadside services (gas stations, car washes, etc), categorized destinations (amusement parks, restaurants), or alternative routes around congested areas, accidents, or construction sites. Many standard GIS systems cannot handle the response-time and multiple user requirements of mobile ATIS clients, so the authors propose a distributed processing and data handling scheme to improve performance.

Choy et al also develop a more ATIS / vehicle navigation-oriented GIS database which differs from the layer and object models. The authors point out that raster layers and linked objects do not represent traffic movement or route connectivity adequately, and routing algorithms extended onto these models cannot distinguish important feature differences (such as overpasses vs intersections) without creating additional object types. The authors describe their desired alternative as a “Naturally Navigable” database.

At the time this article was written, the availability of high-speed mobile data communication was limited, but they still investigate the possibility of a distributed processing network spread over mobile ATIS client systems. They also deal with issues of concurrency and data privacy for multiple users. For the network backbone they propose a central server farm with high and medium-performance systems, linked to regional server locations and client units in a hierarchical caching structure. The servers handle database requests in much the same way as a DNS system, with local data available at the regional level, and global or addressing data available on the central servers.

The authors classify data as either static (relatively unchanging, such as roads and business locations), and dynamic (short-term conditions such as traffic patterns). They utilize separate object-oriented data models for each type, a relational model for static data and a temporal relational model for dynamic information. The temporal model is based on relational database techniques and object definitions as well as a set of temporal functions. An example of a temporal function would be to return an estimate of future traffic density based on past data. Data aggregation is done both spatially and temporally to optimize storage and performance. Spatial aggregation is done by averaging and interpolating data from point samples, such as freeway congestion measurements. Temporal aggregation averages outdated data into generalized statistics so that historical queries can still be performed, but outdated details are not retained. Less-traveled areas retain less detailed data statistics (or average over longer intervals).

Choy et al envision a hierarchical distributed query processing in addition to distributed data storage. Client units forward queries to regional sites if they are overloaded or lack data, and regional sites can forward queries to the central servers. A prioritizing scheme can prevent overloading of the central servers at peak usage times. The authors suggest two operation modes, "normal" providing periodic information updates from clients through regional sites to the central servers. In "emergency" operation, information is exchanged between clients through the central site immediately, such as updates regarding accidents.

Concurrency control is needed for updates on dynamic information (which can change data mid-query), especially if client vehicle units can provide part of the ATIS data (such as speed, accident notification, etc). Concurrency is handled by executing queries as if they were in serial rather than parallel, so the results of one query (such as an alternate route request) can influence the results of a similar simultaneous one (so as not to suggest identical detours to every user on a freeway during rush hour). Serialization is enforced either by two-phase locking, time-based locking, or other methods.

Privacy is enforced through an anonymous vehicle policy which retains important information while dropping identifying data. For network billing, query time and origin can be decoupled from content and results and stored as two separate relation statistics. If better accounting is required by users, then cryptographic techniques can be implemented. (It can be noted here that not all ATIS systems in use today offer strict privacy protection, as users have been fined for speeding based on rental vehicle GPS data).

Choy, Kwan, and Leong go into extensive detail regarding the algorithmic implementation and testing of their ATIS model. They demonstrate the relative efficiency of the system and its robustness in scenarios where regional or even the central servers are not available. The authors note that a more efficient and standardized object-oriented database languages is needed for maximum performance, and that bandwidth limitations also hinder the effectiveness of the ATIS. (This is still true for some areas of the US and many other parts of the world). This article provides an excellent demonstration of the practical applications of GIS data models.

Sengupta and Yan attempt to deal with the storage and querying of spatial information by integrating two separate concepts of field-based dataset handling. By eliminating duplicate spatio-temporal data elements and partitioning space into discrete units, the authors create a a Hybrid Spatio-Temporal Data Model and structure (HST-DMS) (4).

The HST-DMS is an extension of earlier hybrid data models, and stores event-based changes as raster cell arrays for each time period. Linear changes are divided into those which happen prior to the current time step, and those which happen during the current step. Some elements are static and never (or infrequently) change. To reduce the computational complexity that time steps introduce, the data is stored in tree structures which can more efficiently represent change and time overlap.

Testing of this structural design involved transformation of urban sprawl from raster maps into vector polygons, which were stored in both overlapping tree structures and nonoverlapping trees. In the overlapping trees, base maps (static) and change maps (dynamic) were stored as 2D arrays, and a doubly-linked list represented the spatial changes between two times (or temporal states). The overlapping trees improved storage density by saving duplicated raster cells in change maps only once. This proved to be at least twice as efficient in storage requirements for the same data set. Spatial and temporal queries over the data also proved to be faster for the overlapping trees than for other methods.

While this model initially seems to be less evolved than earlier designs (in that it uses raster-based data), it is quite effective at storing certain types of time-based data in this raster/layer form. The HST-DMS model or a derivative would be a good choice for certain application-specific GIS uses.

3. Conclusion

While there is no universally accepted data model for Geographic Information Systems, development in the field has largely focused on the object-oriented spatiotemporal paradigm. With a few exceptions, GIS is moving from a primarily layer-based mapping system to a technology capable of integrating flat maps, dynamically located objects, time-based events and other types of data into a more holistic representation of geographic regions, with the potential for operation in real-time with distributed sensors and mobile users. As the user-base spreads from special-purpose industry and research to include the general public, GIS vendors will be driven to produce more flexible models capable of responding quickly and accurately to a high volume of queries. It remains to be seen whether one of the models investigated here will be extended and developed to become an industry standard, or if an entirely new model will define the future of GIS.

References

1. Choy, M., Kwan, M.-P., and Leong, H. V. (2000) Distributed database design for mobile geographical applications. *Journal of Database Management* Vol 11(1): 3-15.

2. Gunter, O., and Rikert, W. (1993) The design of GODOT: an object-oriented geographical information system. *IEEE Data Engineering Bulle*tin 16: 3.

3. Morris, A. (2003) A framework for modeling uncertainty in spatial databases. *Transactions in GIS* 7(1): 83-101

4. Sengupta, R., and Yan, C. (2004) A hybrid spatio-temporal data model and structure (hst-dms) for efficient storage and retrieval of land use information. *Transactions in GIS* 8(3): 351-366

5. Shekhar, S., Coyle, M., Goyal, B., Liu, D., and Sarkar, S. (1997) Data models in geographic information systems. *Communications of the ACM* 40(4): 103-111

6. Tang, A. Y., Adams, T. M., and Usery, E. L. (1996) A spatial data model for feature-based geographical information systems. *Int. J. Geographical Information Systems*. 10(5): 643-659

7. Worboys, M. and Hornsby, K. (2004) "From objects to events:GEM, the geospatial event model". *Third International Conference on GIScience 2004* M. Engenhofer, C. Freska, and H. Miller (eds.) Lecture Notes in Computer Science 3234, Berlin, SpringerVerlag: 327-344.