GEOGRAPHIC LOGICAL DATABASE MODEL REQUIREMENTS

Martin Feuchtwanger
Department of Surveying Engineering, University of Calgary
2500 University Dr. NW, Calgary, Alberta, T2N 1N4, Canada

ABSTRACT

An important problem of GIS technology is the proper storage and retrieval of geographic data, and the logical database model, i.e. the approach taken in specifying the structure and meaning of and the operations performed on the stored data, is fundamental to its solution. A geographic logical database model should obey all the principles of:
a) syntactic database models, including file integration, controlled redundancy, unified data language, centralized access, independence, abstraction, concurrency, distribution, integrity and security; b) semantic database models, including abstract objects, relationship types, object-oriented query, knowledge incorporation, relativism and evolvability; and c) geographic data processing, including taxonomy, temporality, symbology, geometry, uncertainty, theme integration and view generalization. When such a model is provided, the design of geographic databases for sophisticated GISs will be possible.

INTRODUCTION

The purpose of this paper is to outline the requirements of a logical database model suitable for GIS.

A database is an organized set of interrelated data and is the central component of any information system. The design of the structure of a database is known as a schema and it is specified using a data language. The software package that defines the structure of and handles all access to a database is termed a database management system (DBMS).

What distinguishes different kinds of DBMS is an underlying theory known as a database model. It can be defined as a conceptual tool for describing data (or entity) types, their relationships, operations and constraints. Different classes of data models vary according to how closely they are oriented toward human or machine understanding, respectively. Logical database models, e.g. the entity-relationship, semantic, relational and network data models, help us specify what is going on. Physical database models relate to how the logical specifications are implemented and are not further discussed.

Consider the following: a geographic database (GDB) is a database appropriate for a GIS; a geographic DBMS (GDBMS) is a special DBMS controlling the nature and content of a GDB; and a geographic database model is a theory guiding the design of a GDBMS. What sort of logical database model do we want for GIS? One that will enable us to build, maintain and use sophisticated geographic databases. The model has a

number of important requirements. Some are specifically geographic and others apply to all information systems.

Although this paper is nominally practical, for the design of geographic databases, it could claim to move toward a general theory of geographic information.

MODEL TYPES and USERS and DATABASE VIEWS and DOMAINS

Four questions help guide the rest of the discussion.

What are the Main Types of Database Model?

Logical database models themselves can be subdivided according to their levels of abstraction.

The earlier ones are relatively low-level and are called datalogical (or syntactic) data models. They are also referred to as record-based because the entities involved are files, records, links and fields.

Newer database models are relatively high-level and are described as being infological (or semantic). They are also known as object-based because the entities involved are sets, objects, relationships and attributes.

Who Uses a Database Model?

The users of the database model can be considered to be database designers, application programmers or application programs, i.e. people or software, the distinction does not matter for this discussion. Strictly, they do not use the model directly, they use a DBMS, but again the distinction does not matter most of the time. End-user issues, such as "natural" query languages are not discussed.

What are the Major Views of a Geographic Database?

A view is a logical component of a database, as seen by a particular user. It is not necessarily all that exists.

The first major view of the database is that it is a real world simulation, an environmental model. It contains all of the data required for most purposes and may be called the phenomenal view because it describes the phenomena of interest.

The second major view of the database is that it is a symbolic (or visual) representation of the real world. It is derived from the phenomenal view by some design (or rendering) process and it may be called the cartographic view because it describes essentially a map.

Both views are digital and both exist at different levels of generalization (or scales).

What are the Major Domains of a Geographic Database?

Within each view (or portion) of a database are stored numerous kinds of data. Each datum (or piece of data) can

be considered to be drawn from a general domain, a class of data types, and each describes a different characteristic (or attribute) of an object or concept being represented in the database. Five major categories are identified.

Taxonomic (or thematic) data on an object tell us what it is. Examples include names, classes and values. They are often labeled simply as "attributes" although here the word refers to the more general concept.

Temporal (or historical) data, e.g. epochs and periods, on an object tell us when it is or was.

Symbolic (or visual) data, e.g. annotation, colors and shadings, on an object tell us what it looks like on maps.

Geometric (or spatial) data on an object tell us where and what shape it is. They can be either metrical or topological and include locations, coordinates and neighbors.

Scientific (or theoretical) data on an object tell us why it is, and may include laws, explanations and production rules.

Each domain is relevant to both the phenomenal and the cartographic views, except the symbolic domain which applies only to the cartographic view.

GENERAL REQUIREMENTS

The requirements of general-purpose database models are grouped according to how closely they relate to the world of the application environment relative to the workings of computer systems.

Characteristics of Standard Record-based Data Models

A geographic DBMS should at least exhibit all the characteristics of a conventional DBMS [Frank 1988]. Details can be found in most standard database texts. Briefly they are:

Integration. Several separate but related files are somehow combined into one unified whole, a database.

Controlled redundancy. The same data are not being duplicated, at least not unnecessarily or in such a way that inconsistency is possible.

Unified data language. A means of specifying both the structure of and the atomic operations to be performed on a database is provided.

Centralized access. Only one means of accessing the data, a common and controlled one, is used by all.

Independence. Programs are independent of the way data are stored, and data are independent of the way programs are implemented. Changes to one will not affect the logical characteristics of the other.

Abstraction. The same data exist at (or are viewed at) different levels of abstraction. Details of data viewed at one level are hidden from those at the next higher level.

Concurrency. Several users may access the same data at the same time.

Distribution. One large data set is stored or made available at several different sites.

Integrity. There is protection against illegal operations being performed on certain objects and against inappropriate objects being operated on by certain procedures.

Security. Data are protected against unauthorized access and against loss due to system failure.

The problem with such database models for any sophisticated information system is their limited semantic expressiveness.

SEMANTIC EXPRESSIVENESS of OBJECT-BASED DATA MODELS

A geographic DBMS must perform much more of the work that is required to properly manage GIS data than is currently the case. Much burden is placed on the application programs to perform many data management tasks, such as integrity maintenance. That leads to duplication of effort and program-data dependence, both contrary to the goals of database systems. Users of a GDB must have a facility for expressing concepts meaningful and useful to an application environment without excessive use of programs. They require data models with more semantic power.

Data semantics are the meaning, structural properties (i.e. objects and relationships), operational characteristics and integrity constraints of data [King & McLeod 1985; Su 1986]. Ideally, semantic data accurately reflect real-world objects or concepts. Following are some of the major requirements for achieving semantic expressiveness in database models.

Structures

The structure of a database, also called database statics, include various different object and relationship types.

Objects. A data model must provide a set of useful generic data types. As well as character, integer and real, for example, vector and tessellation could be provided. Also, it must be possible to combine the basic types into more complex ones, to suit particular applications. Ideally, there must be a simple, direct correspondence between an entity in the real world and an entity held in the database. Such is the notion of the abstract object [King & McLeod 1985]. Objects should be allowed to represent themselves directly instead of using some artificial identifier. Users should not have to be concerned with polygon IDs, for example, if such codes have no meaning in real life.

Relationships. The model must explicitly provide for different kinds of relationships (or associations) between entities [King & McLeod 1985; Su 1986]. Three are fundamental.

When an entity represents some action, mapping or relationship between two or more entities there is said to be an interaction type of relationship. For example, a land ownership represents a mapping between a land parcel and a land owner, and two roads may interact at an intersection.

When one or more entities together describe (or are attributes of) another usually more complex entity, there is an aggregation type of relationship. For example, a set of districts may form a region, and a name, a location and a population may describe a city.

When one or more entities are each subtypes of another more general entity there is said to be a generalization. For example, fir, spruce and pine are types of conifer, and transportation line may be a superclass for highway, railway and waterway.

Operations

The allowed operations (or transactions) on a database are also known as database dynamics. Basic database operations include retrieval, printing, deletion and insertion of an object based on certain attribute value conditions. Other ones involve whole sets of objects. Advanced operations might include certain display or mathematical functions on objects.

Expressing procedures as a series of basic operations is said to be navigational, in which case a high-level operation must be formulated using many query steps. Even with some non-navigational languages, where most operations can be specified with single queries, an intimate knowledge of the data structure is still required. In which case, a high-level operation must be formulated using a very complex query.

A user should not be burdened with using such a complex data language. A so-called object-oriented query mechanism should be provided whereby high-level transactions can be specified very simply.

Knowledge

If a GIS is going to be a decision-support system, produce well-designed maps, maintain a high degree of integrity, or live up to any more of its many expectations, it will have to be founded not just upon a database but upon a knowledge-base [Karimi & Feuchtwanger 1989].

A knowledge-base can be considered to be a database, extended to incorporate knowledge concepts. As well as the data (facts or assertions) that are in a database, it has expert rules for inferring new facts or rules from existing ones. Rule types include production rules, and semantic and security integrity constraints. They may be general or application specific. Analogous to the DBMS will be a knowledge-base management system (KBMS). It has an inference mechanism for applying the stored rules and an ability to explain the knowledge-base structure.

Thus, the data model must provide for the incorporation and use of knowledge into the schema by making the knowledge definable along with the objects to which it applies.

Relativism

Any GDB is likely to have several different classes of user, each one having different assumptions or expectations about the structure, operations and contents of the database. Each user may see the same object in a different way and have quite different authority regarding retrieval, update or analysis operations. Take the case of a lot. Different attributes of an entity "parcel" are of interest to different users --

surveyors: lot boundary measurements and owner;
planners: census data of its inhabitants;
engineers: location and specifications of its utilities.

Also, "owner" may be seen as just an attribute of an entity "lot" to one user but to another it is seen as an independent entity "person" joined by a relationship "owns."

One ought to be able to conceptualize different views of the same information as a semantic unit [King & McLeod 1985]. The model must have the ability to allow alternate views of and authority over a data set by different users. Such a concept is known as relativism.

Evolvability

A comprehensive GIS involves so many different data themes, accessed by many different users, at several sites that it is unlikely that any early database design is going to be adequate for all subsequent applications. Once a schema has been set and the database has been populated, it should be possible to modify the schema without the contents being corrupted. A model must possess evolvability, i.e. have the ability to allow a schema to evolve with changing knowledge or specifications of the application environment [King & McLeod 1985].

GEOGRAPHIC REQUIREMENTS

Geographic information systems are generally larger and more complex than most other types of information system. They have a number of significant logical database modelling requirements that are additional to those presented above. A set of particularly geographic objects, attributes, relationships, operations and constraints must be provided by the model. Space does not permit a full coverage of such constructs, so only a few examples appear below. The requirements discussed are those relating to the taxonomic, temporal, symbolic and geometric domains, presented above, plus those concerning data uncertainty, layer integration and levels of generalization.

Taxonomy

Conventional hierarchical classification schemes may be inadequate for complex GIS databases. Multi-branched tree structures with super object types, object types and sub-object types are called generalization hierarchies when represented using a logical data model. They can be too simple or restrictive, however, when an object is considered

to belong to more than one immediate class of objects. A river, for example, may simultaneously belong to the following classes: transport route, national boundary, drainage channel and waterbody. To model such situations a generalization network is required [Su et al. 1988]. More examples of generic objects that may be required in the taxonomic portion of the model include: thematic layer, categorical coverage, linear network, region and pointal feature.

Temporality

As the environment changes, so too must the database by which it is modelled, if the GIS is to maintain usefulness. However, a historical record is often useful too. As well as the spatial domain that commonly characterizes a GIS, there must be a temporal domain to the model. That is, the model must be capable of handling the monitoring of what changes take place, where and when they occur [Langran & Chrisman 1988]. It should contain data types such as date and period and should facilitate the answering of questions such as "what was the condition at location *X* in 1986?" and "when did the condition at location *X* change to situation *S*?" or certain kinds of time series analysis.

Symbology

Within a GDB, at any given scale, there must be a single view of the phenomena and possibly several corresponding views that are cartographic. The phenomenal view would be independent of graphic symbology. The cartographic views (or maps) contain their own symbology and would be derived from the phenomenal view by a map design (or rendering) process. That way, important data on any aspect of the phenomena and any kinds of analysis done on them are kept logically separate from the way they are visually represented. Any visualization of the phenomena is likely to contain only a partial subset of the phenomenal view and must involve a sophisticated cartographic design process, if it is to be optimal.

The model must therefore exhibit what may be called phenomena-cartography independence, the former being independent and the latter being dependent. Generic cartographic object type examples might include: line symbol, point symbol, label, legend and title. A simple cartographic operation type is "display an object" while a very high-level example would be "design a map."

Geometry

For any given geographic object (in either the phenomenal or the cartographic view) there may be more than one alternative geometric structure used to spatially represent it. Many have been proposed over the years and because different structures are optimal for different purposes more than one may be desired, even for the same object.

A single class of geometric objects may be used to represent many different kinds of geographic object, depending on the

application. For example, a polygon can represent a forest stand, a geological outcrop, a cadastral lot or a terrain patch. Also, several application objects may be simultaneously represented by the same instance of a geometric object. For example, several linear features may be spatially represented by the same polyline. However, the application level user must not be aware of (or be encumbered by) the particulars of the raster/vector implementation.

Thus, the model must support a level of abstraction at which geographic concepts are expressed independently of the geometry used to represent them [Feuchtwanger 1985]. The concept may be called application-geometry independence, the application level being independent of the geometric level. The latter contains many different spatial object, relationship, operation and constraint types.

Geometric object examples include: 0-cell (or point), 1-cell (or line), 2-cell (or polygon), 3-cell (or patch), tessellation (or image), run-length-coded bitmap, region quadtree, etc. Spatial relationships include: adjacency (or primary neighbor), proximity (or secondary neighbor), exhaustive subdivision, discontiguous homogeneous group, etc. Spatial retrieval operations include: the objects overlapping an object, the nearest object to a point, the subregions constituting a region, the geographic objects represented by a geometric object, etc. Geometrical integrity constraints include shape preservation and topological consistency [Mepham 1989; Zhang 1989].

Uncertainty

All taxonomic, temporal and spatial data are uncertain to some degree, whether from phenomenal fuzziness, measurement error or machine imprecision [Miller et al. 1989]. Categories of uncertainty include accuracy, precision and resolution. The uncertainty associated with (or quality of) all geographic data must be accounted for if the GIS is going to be credible. Also, the propagation of uncertainty during data processing and data combination must be handled. The model must have explicit facilities for associating uncertainty attributes to data and for appropriately combining uncertainties during data operations.

Integration

In a GIS there will be several different themes (or layers) of data relating to the same region. For many analysis applications, the integration (or overlay) of two or more of these layers will be required. Relationships between different layers can be explicitly represented within the database and retrieved when required, or derived computationally (either by the DBMS or by application software) when required. Since space is taken to store them, or time is taken to compute them, care is needed during design. For other applications, the different layers are otherwise quite unrelated and should be kept apart.

Thus, the model must allow for different degrees of

permanent integration to be specified. The concept could be called variable integration where the relationships between elements of different layers may be all, partly or not stored. An example of permanent integration is when the overlay of two polygon networks yields a composite network of common, smaller polygons.

Generalization

The ability to model the environment at different levels of generalization (or detail) is an essential characteristic of any sophisticated GIS. All views of reality are subjectively dependent on the scale of investigation and it is important that the model facilitates views of differing levels of generalization. Strictly, each level is dependent on its more detailed level and cannot be updated except via a special process, the generalization process.

For example, a small-scale route map might be derived from a large-scale topographic map. If a new road is built, the topography is updated and then appropriate data is propagated to the route map. A complete GIS may have a series of different phenomenal views, with only the most detailed one being updatable from the outside. The concept might be called multiple generalization and seems to contrast with the myth of the scale-free database. Generalization operations range from the low-level line simplification algorithms to the complex generalization of an entire thematic layer.

PRACTICAL CONSIDERATIONS

Although the model is a conceptual tool there are a number of practical things to consider when developing it.

Usable. It must be possible to use the model, i.e. to build suitable GIS schemas and to easily specify the storage and retrieval of geographic objects. It would be possible by means of the data language that must accompany the model.

Implementable. The model must not remain only theoretical; it must be implementable in the form of a GDBMS. That is the physical database design problem. How it is implemented raises many more questions that are beyond the scope of this paper. Briefly, the GDBMS may exist as an extension to some existing DBMS or as a completely redesigned package. Either way, special attention must be paid to efficient spatial access [Frank 1988].

Modifiable. The development of a logical database model for GIS is both a new and ongoing project. For the present, the model should be in a state of flux, not be fixed in stone; there must be room for modification, expansion or improvement.

General. From a computing point-of-view, the model must be more special-purpose than general-purpose, i.e. it facilitates geographic database design and use. From a geographic point-of-view, it must be general and simple

enough to be useful for most purposes, i.e. it is not so special or unduly complicated that it excludes certain applications. Finding a happy medium might be called the model designer's generality problem.

CONCLUSION

To conclude, a summary of the requirements of, implications of the existence of and recommendations for providing a geographic semantic database model are given.

Summary

A geographic logical database model should obey all the principles of: conventional database models, general semantic database models and geographic data processing.

Standard database principles include file integration, controlled redundancy, unified data language, centralized access, program-data independence, levels of abstraction, access concurrency, and data distribution, integrity and security. Semantic database principles include abstract objects, relationship types, object-oriented query, knowledge incorporation, relativism and evolvability.

Additional GIS principles involve: taxonomy (i.e. the generalization network), temporality, symbology (i.e. phenomena-cartography independence), geometry (i.e. application-geometry independence), uncertainty, integration (i.e. variable integration) and generalization (i.e. multiple generalization).

Implications

Current database models are inadequate even for general information systems. The relational model, for example, does describe structures and operations but has very limited semantics. Current semantic database models may also be inadequate for GIS. The entity-relationship model, one of the first semantic models, only describes database statics not dynamics. Neither kind expressly handles geometrically based applications.

A model exhibiting all of the specifications outlined above may be a long time in coming, but it will facilitate the design and use of sophisticated GDBs because it will have a much closer association with concepts in the world of geography.

Recommendations

Many of the above problems have been individually attacked by other researchers, but an integrated approach is needed. Also, much work is still to be done on developing the model. In particular, a comprehensive, integrated and formal set of geographic semantics should be produced and existing semantic data models should be investigated to see how well they cope with the added geographic requirements.

REFERENCES

Feuchtwanger, Martin, 1985, An Investigation of Efficient Computer Techniques for the Storage and Retrieval of Land-related Information, M.Sc. Thesis, Department of Surveying Engineering, University of Calgary, Alberta, Pub. no. 20010.

Frank, Andrew U., 1988, "Requirements for a Database Management System for a GIS," PE&RS, 54:11, 1557-1564.

Karimi, H.A. and M. Feuchtwanger, 1989, "Geographic Knowledge Base Management System (GKBMS): The Future Challenge in Geomatics," presented paper, National GIS Conference, Ottawa, February/March.

King, Roger and Dennis McLeod, 1985, "Semantic Data Models," in Principles of Database Design, Volume I, Logical Organizations, S. Bing Yao, ed, Englewood-Cliffs, New Jersey: Prentice-Hall, 115-150.

Langran, Gail and Nicholas R. Chrisman, 1988, "A Framework for Temporal Geographic Information," unpub. paper, Department of Geography, University of Washington, Seattle.

Mepham, Michael P., 1989, Automated Cadastral Data Entry Into an LIS, unpub. Ph.D. Thesis, Department of Surveying Engineering, University of Calgary, Alberta.

Miller, R., H.A. Karimi and M. Feuchtwanger, 1989, "Uncertainty and its Management in Geographic Information Systems," presented paper, National GIS Conference, Ottawa, February/March.

Su, Stanley Y.W., 1986, "Modeling Integrated Manufacturing Data with SAM*," IEEE Computer, 19:1, 34-49.

Su, Stanley Y.W., Vishu Krishnamurthy and Herman Lam, 1988, "An Object-oriented Semantic Association Model (OSAM*)," in AI in Industrial Engineering and Manufacturing: Theoretical Issues and Applications, Kumara et al. eds, American Institute of Industrial Engineers.

Zhang, Guangyu, 1989, Consistency Preserving Transactions for Automated Cadastral Database Systems, unpub. M.Sc. Thesis, Department of Surveying Engineering, University of Calgary, Alberta.