ATTRIBUTE HANDLING FOR GEOGRAPHIC INFORMATION SYSTEMS

Peter Aronson
Environmental Systems Research Institute
380 New York Street
Redlands, CA 92373

ABSTRACT

Geographic information systems manipulate and manage both spatial information and the thematic attributes of that information. There are several candidate methodologies for the management of these thematic attributes for system designer to choose among. Which is the most useful, both in terms of data model and of normal usage? This paper discusses the choices open to the system designer in the context of both sets of criteria.

1 INTRODUCTION

Real world geographic entities can be modeled in a Geographic Information System (GIS) as features composed of a set of locational information (position, geometry and topology) and a set of thematic information. The handling of locational information is beginning to be somewhat understood, and there exist widely accepted paradigms to deal with it (such as the topological model). The handling of thematic data, of sets of attributes, while well understood in general, is not well understood in terms of GIS processing.

The subject of thematic data handling has been very well studied in general however. Database Management Systems (DBMS) have been in use for well over twenty years, and in that time many advances have been made. Modern DBMS manage data using sophisticated techniques drawn from various branches of mathematics (such as set theory and graph theory) as well as the latest techniques of computer science. Several of these techniques have been incorporated into existing GIS.

The approach used to manage thematic data can not be examined independently of the GIS data processing model upon which it is based. A GIS a is geographic database and a set of operations upon that database - the form of operations performed on the spatial portion of that database specifies the form of operations required upon the thematic portion.

While the data models and techniques used to manipulate thematic data are important, equally or more important are the organizational procedures involved in that data's collection, evaluation and use. Organizations that produce and use data have needs distinct and separate from the requirements of the software. The organization is not going to change, so a GIS's thematic data handling must be able to match that organization's needs, or it will not be used.

This paper is organized into five sections: an introduction into the nature of the problems involved in thematic data handling for GIS; a survey of thematic data models currently in use; a discussion of GIS processing models and their implications for associated thematic data processing; a discussion of the organizational constraints on the handling of thematic data; and finally, conclusions on the above.

2 THEMATIC DATA MODELS

There are many different paradigms for the management of thematic data. The most common are: Tabular; Hierarchical; Network; Relational; and Object-Oriented. The first is the manner in which most early GIS stored their attribute data (if any), the next three are those currently most commonly implemented in DBMS, while the last is newer but rapidly gaining in popularity for some applications.

The simple tabular model sees data as collections of independent tables with rows (records) and columns (fields). These usually will have fixed field definitions, but aren't required to. Fields may be variable length or repeating. Such systems will usually have simple query systems if at all.

2.1 Simple Tabular Model

The simple tabular model allows the association of attribute codes with geographic features. Its major lacks are in terms of data integrity (since each table is independent, identical data to be used with two different tables must be present in both, which means they can disagree), storage efficiency, and flexibility; however such data structures are easy to program and to convert from system to system.

2.2 Hierarchical Model

A hierarchical database organizes data in a tree structure. A tree is composed of a hierarchy of elements. The uppermost level of the hierarchy has only one element, the root. With the exception of this root, every element has one element related to it at a higher level, referred to as its parent. No element can have more than one parent. Each element can have one or more other elements related to it at a lower level, referred to as that element's children (Martin, 1975).

Hierarchical DBMS have not gained any noticeble acceptance for use in GIS. They are oriented for data sets that are very stable, where primary relationships among the data change infrequently or never at all, since the data relationships are built into the logical view of the database. Also, the limitation on the number of parents that a element may have is not always found in actual geographic data (the section of US Highway 215 immediately south of US 10 would have two parents in the California Highway database, US 215 and California 91). Finally, the query language

for a hierarchical DBMS is of necessity procedural, that is, it requires knowledge by the user of the actual storage scheme used by the DBMS. This is information that the user should definitely not be required to know.

2.3 Network Model

A network database organizes data in a network or plex structure. Any item in a plex structure can be linked to any other. Like a tree stucture, a plex structure can be described in terms of children and parents. In a plex structure, children may have more than one parent, and link back upwards (that is, an element can be its own grandparent or even parent) (Martin, 1975).

Network DBMS have not found much more acceptance in GIS than hierarchical DBMS, although they are not without their champions (Frank, 1982). They have the same flexibility limitations as hierarchial databases; however, the more powerful structure for representing data relationships allows a better modelling of the data relationships found in geographic data. The query language, however, for network databases is still procedural.

2.4 Relational Model

In a relational database, information is organized in tables. These tables have a more rigorous definition that in the simple tabular model. The tables, which are identified by unique table names, are organized by column and row. Each column within a table has a unique name. The set of values from which the actual values in a column are drawn is called the domain of the column, and may be shared among different columns (within different tables). Each row (or tuple) is a set of permanantly associated values. Tables may be joined to each other by columns sharing a common domain. Such joins are usually ad hoc and temporary operations, unlike the previously discussed database types, these relationships are implicit in the character of the data as opposed to explicit characteristics of the database set up. A simple example of a join of two tables in a relational database:

SOIL_POLYGONS

Poly#	Area	Soil_Code
1	37.5	AN 32
2	15.6	CE 12
3	41.7	BG 17
4	22.1	AN 32

SOIL_DATA

Soil_Code	PH	Sample_Date
BG 17	-1.7	11/23/84
CE 12	+3.2	04/06/82
AN 32	+1.7	12/22/81

Since both Soil_Code columns share the same domain (legal soil type identifiers), the two tables can be joined by soil code. This

yields the resulting table:

SOIL_POLYGONS + SOIL_DATA

Poly#	Area	Soil_Code	PH	Sample_Date
1	37.5	AN 32	+1.7	12/22/81
2	15.6	CE 12	+3.2	04/06/82
3	41.7	BG 17	-1.7	11/23/84
4	22.1	AN 32	+1.7	12/22/81

Note this result need not be an actual table, but can be generated as required. This results in a smaller storage requirement (there is no redundent storage of information for soil AN 32), and a more normalized data structure (see below). Note, a different result could be produced by joining the Soil_Polygon table with yet another table, say a Polygon_Symbology table, joined by the Poly# domain.

The relational database model is the most widely accepted for managing the attributes of geographic data, examples including SGIS, GEOVIEW (Waugh & Healy, 1986) and, ARC/INFO (Morehouse, 1985). It is attractive because of its simplicity (all data stored in simple tables), its flexiblity (any set of tables can be joined together by columns with common domains), efficiency of storage (by proper design of tables, redundant information can be eliminated) and by its non-procedural nature (queries on a relational database do not need to take into account the internal organization of the data). The relational DBMS has emerged as the dominant commercial data management tool of the eighties.

2.5 Object-Oriented Model

The basic unit that an object-oriented DBMS manages is the object. An object is a collection of data elements and operations that together are considered a single entity. Objects are typed, and the format and operations of an object instance are the same as some object prototype. Objects may be hierarchical, that is, objects may be composed of other objects in turn (Wiederhold, 1986). An example of a object might be a swamp object:

Swamp Object:

List of Border Chains: C1, C2, C3,...,Cn

List of Nodes: N1, N2, N3,...,Nn

Attributes: Depth
Muck type

Soil Samples: S1,...,Sn

Symbology: Solid borders
blue shade
random swamp symbols

Operations: Measure
Drain
Expand

Once this structure is set up, the details of it need not be user visible. The above is a relatively passive view of an item. In some systems objects take a very dynamic role, being the primary means for rules to be implemented.

As noted above, the object-oriented database is a relatively new model, although its origins go back to work done at Xerox in the early seventies. So far, the only geographic data handling system to extensively employ this model is TIGRIS (Wientzen, 1986). This approach has the attraction that query is very natural, as features can be bundled together with attributes at the database administrator's discretion. It is however considerably less ad hoc than the relational model, and is not normalized.

In addition to the above pure systems composite systems exist as well that combine characteristics of two or more models, such as relational-hierarchical or object-oriented-relational.

3 GIS PROCESSING MODELS

In general, the form of thematic attribute processing appropriate for a GIS depends on the data processing model that it uses. A data processing model is a formalization of operations on data, as opposed to a data representation model, which is a formalization of a real world object or structure (an example of a GIS data representation model is the USGS DLG format, which is a formal model of a USGS topographic quad sheet).

In the context of this discussion, map processing will be discussed independent of the data structures and algorithms involved. In these terms polygon overlay and grid cell overlay are the same operation - spatial join. Only the operations are considerered, not the algorithms nor the representation of the maps themselves. There are three such models commonly used for mapping: the simple map; the composite map; and the relational map.

3.1 Simple Map Processing Model

The simple map processing model assumes that a data set represents a single map sheet. Each data set is thematically atomic, that is, it can not be split into multiple maps by subject - there is only one or no sets of attributes per data set. All attributes are

tightly bound to the map; there is no thematic data available except that one set. And it is thematically independent - data sets can not be combined. Examples of simple map models are CAD/CAM systems or many of the simple mapping packages (such as SYMAP).

The simple map model, if it has any thematic data handling at all, uses the simple tabular approach or something functionally equivalent to it, since there is no access to other thematic data or map data sets. There is no need for systems that can handle combined or linked data sets as they never occur.

A pure contouring package would be an example of the simple map processing model. All operations occur upon one set of data (points) and involve only that set of data and its attributes (elevations). There exists no mechanism for joining two data sets by spatial domain (locations) and all operations involve only one data set at a time. The operations are all in the form of F(ds1) -> ds2 (where F is a function on a data set such as rescaling, and ds1 and ds2 are data sets) or the form F(ds1) -> Vds1 (where Vds1 is a "virtual" data set on the order of Moellering's virtual map (Moellering, 1984), and F is an function on the data set such as contouring where the output is a graphic or report). Of neccessity, the types of operations on the thematic data is limited to the types of operations on data sets as a whole.

3.2 The Composite Map Processing Model

The composite map processing model assumes that a data set is a combined set of map sheets. If you add to the simple map model the operation of spatial joining, of overlay, the result is the simplest form of the composite map processing model. In the composite map model, because spatial joins can have occurred, there will be N sets of attributes for each data set, where N is the number of source map sheets that contributed to the data set. The thematic data available for a data set is the sum of all the original map sheets. Operations using the composite map model occur within an assembled data set - combination occurs before further processing.

Attribute handling for this composite processing model can take one of two basic approaches. Once again the simple tabular model can be applied, in which case during the construction of the composite data set, a composite attribute table is also constructed. Attribute operations then occur on this table. Operations occur on the resulting composite spatial elements. The alternative approach is to classify the results of the combination into objects. These objects reassemble the original pre-combination features out of the smaller post-combination elements. These objects contain as a result all the combinations of data overlayed by the resulting object as it is pointed to by the post-combination

elements. This allows data to aggregated as needed.

An example of the composite map processing model is a simple GIS with overlay capabilty such as GRID (Tomlinson, et al, 1976). Operations still occur on only one set of data, with the exception of one particular operation, the overlay. After an overlay (an operation of the form F(ds1,ds2) -> ds3) there is a thematic data set with combining the thematic data from all input data sets, permanantly joined by common spatial domain, that is, by the common resulting grid cell. Query and reporting operations can now operate on this composite data set, performing such operations as identifying cells that have value A in ds1 and value B in ds2. A more sophisticated system might be able to identify features that are borders between value A is ds1 and B and in ds2 and have value C is ds3. In all of these cases, operations can only happen on data that has been built into the composite data set.

3.3 The Relational Map Processing Model

The relational map processing model looks at a data set as a set of spatially overlapping, independent map sheets and associated attribute tables. These map sheets are combinable but not permanantly combined. Each map sheet represents a normalized relation consisting of a spatial key (location) and a set of attribute tables. Operations within the relational processing model occur ad hoc as needed between independent elements of the the data set. Also, unlike the above two models, the data set is not sharply bounded - any available data in the proper format may be included in an operation with any other data (assuming they share either a spatial or a thematic domain).

The obvious data management model for the relational map processing model is relational, since it is essentially an extension of the relational model by the addition of spatial joins (overlays). That is, both deal with data sets that can be joined on common domains as required. A useful extention to this model is to allow these joins to occur across multiple DBMSs.

Within this data model, ideally each attribute table, whether attached to a map sheet or not, should be in at least third normal form (3NF). A table is in 3NF if every determinant is a candidate key (Date, 1975). A determinant is an attribute upon which another attribute is functionally dependent, such as PH is functionally dependent upon Soil_Code in the SOIL_DATA table above. A candidate key is a column or a set of columns whose attribute values uniquely identify all the rows in the table. Even more desirable is a further degree of normalization, fourth normal form (4NF) which requires a further degree of independence. What this means in functional terms is that all of the data in a single table should deal with different aspects of a single subject. This is very important for updating that data (see below for discussion).

A partial example of the relational map processing model is the ARC/INFO GIS (Morehouse, 1985). While operations can occur in single data sets as in the simple map processing model or in combined data sets as in the composite map processing model, there is a third fashion in which operations can occur in the relational map model. That is across two or more independent data sets. In theory this operates much like operations in a relational data base. The user specifies a series of spatial and thematic joins and subsetting objections to create a virtual data set (called a view in a relational DBMS), then operates on this virtual data set as if it was physically existent. The virtual data set would never exist as an actual data set. In practice, spatial joins are difficult enough and costly enough so that they are not practical to perform in an ad hoc manner. The technique used in ARC/INFO for relational map processing is to perform overlays on data sets containing no direct thematic data, but simply pointers to other tables containing it. The data sets resulting from this operation act as indices to allow relations between separate data sets.

An example of this would be to take three map data sets; a soils map, a land use map, and a vegetation map; and four associate thematic tables; soils data, land use data, lease data, and vegetation data. Relations are as follows: soil map -> soil data, land use map -> land use data -> lease data, and vegetation map -> vegetation data. The three maps data sets would be overlayed, producing a map data set that had pointers to three thematic data sets (soils, land use and vegetation). The relational database would then be used to link these five data sets (the four thematic data sets and the index data set) together to answer such queries as "Find those polygons that have arable land, no protected species, and are owned by the state".

It should be noted, that as in the thematic data models, the GIS processing models can also exist in hybrid form. There exist GIS that essentially employ the composite map processing model, but have limited relational capability. To even further confuse attempts at classification it is possible to use a system that employs the composite map processing model as if it used the simple map processing model, or to employ a system that uses the relational map processing model as if it used the composite map processing model. Sophisticated capabilities tend to be ignored by users who don't need them.

4 ORGANIZATIONAL CONSIDERATIONS

Organizations acquire geographic information systems to meet their needs - not the other way around . To be successful a GIS must be able to support the organization's existing internal structure. Attempts to change this will typically run into massive bureaucratic inertia, particularly if the current structure is functioning in a satisfactory fashion.

In most government organizations involved in using public land records, as well as in large corporations that collect map data for their own use (such as oil companies), not all the map data for a region is collected by the same agency. In fact, typically, map data will be collected and maintained by a combination of Federal, State and Local agencies. In the Dane County, Wisconson example described by Chrisman (Chrisman and Niemann, 1985), the seven layers in the database were provided by five organization, two federal, one state and two county. This is typical of land records information in this country. (In the commercial sector the situation can be even more complicated since data is often purchased from multiple service bureaus.)

To make matters more complicated, most agencies usually will have begun automation of their thematic data well before obtaining a GIS. This means that the data will be stored in some DBMS system or another. Often conversion to the GISs own format is undesirable or impractical (such as when the DBMS has capabilities that the thematic data handler for the GIS lacks, such as concurrent access control or a powerful report generator). This situation often leads to the requirement that the system handle thematic data in multiple DBMS.

Not only is data typically provided by an assortment of agencies in a variety of forms, it will usually be maintained by the providing agencies. That data will need to be updated, often frequently. It is here that a normalized database pays off. In a properly normalized database, the data sets (tables and maps) are kept divided into elements that only contain data on one subject, and hence, only from one source agency. Since these data sets are not combined until required, each agency can update its data when needed, without worrying about modifying another agency's data. A virtual data set generated at a later time would then automatically incorporate the latest data. This can be particularly important with certain types of thematic data that are updated so frequently as to require transactional capability in the DBMS that stores it, such as statewide land ownership.

5 CONCLUSIONS

The GIS implementor (by which is meant either someone designing a new GIS or someone integrating an existing system into an organization's operations) has not only the task of modeling some portion of the real world for an organization, but of doing so in a manner supportive of the organization's internal structure. Since data is not typcially collected or even processed by a single, centralized source, this requires the processing of thematic data as independent data sets that can be combined as needed. The primary existing tool for this task is the relational DBMS, and the most practical environment in which to apply it is in a GIS that implements the relational map processing model. Current and future GIS systems would do well to work towards this goal.

REFERENCES

Chrisman N. and Niemann, B., Alternative Routes to a Multipurpose Cadastre: Merging Institutional and Technical Reasoning, Proc. AutoCarto 7, 1985. p. 84-93.

Martin, J. 1975, Computer Data-Base Organization, Prentice-Hall, New Jersey.

Moellering, H., Real maps, virtual maps, and interactive cartography: Spatial Statistics and Models, G. Gaile and C. Willmott (eds).: Boston, Mass, D. Reidel, 1984. p. 109-132.

Morehouse, S., ARC/INFO: A Geo-Relational Model for Spatial Information: Proc. AutoCarto 7, 1985.p. 388-397.

Tomlinson, R., Calkins, H. and Marble, D., Computer Handling of Geographic Data: The UNESCO Press, 1975.

Waugh, T. and Healy, R., The GEOVIEW Design: A Relational Database Appraoch to Geographic Data Handling: Proc. 2nd Intl. Symposium on Spatial Data Handling, 1986. p. 193-212.

Wiederhold, G., Views, Objects, and Databases: Computer, Dec. 1986, p. 37-44.

Wientzen, B., TIGRIS.. An Object-Oriented Approach to Topology: InterVue, 4th Qtr., 1986.