MODELING THE EARTH: A SHORT HISTORY

Michael F. Goodchild[1]

ABSTRACT

Models are partial mirrors, reflecting how humans perceive the world around them. I trace the history of modeling the geographic world from early description and representation in the form of drawings and stick maps to today's geographic information systems. Representations have grown more complex and detailed with time, but no less partial in the sense that their content tells only part of the story, and never tells any part exactly. I argue that representation is cyclical: an approach is adopted in response to some kind of technological advance; it is found to be partially successful, and is adapted and stretched to include phenomena for which it was not designed; and finally as stresses build a new approach is adopted that exploits advances in technology. I examine the current situation in this context, and argue that the cycle will repeat itself before long.

INTRODUCTION

Mapping is almost as ancient as humanity itself, and so also are the terms that people use to describe various aspects of the Earth's surface—nouns and adjectives for features, prepositions to express relationships, and verbs to describe change. A variety of tools have been developed to aid in description, and to create representations that can be stored, shared, and changed as knowledge advances. Among the earliest of these were the sticks used to etch map-like diagrams on mud floors and the sticks and string used by Pacific islanders to aid in navigation. By the time of the invention of the printing press the idea of a map had become codified, along with the concepts of projection and symbolization that were necessary to reduce knowledge of the Earth's surface to a collection of marks on flat paper. With a pen and paper, a cartographer could sketch the outlines of continents, add rivers and roads, and annotate with feature names. The continuous variation of topography presented more of a problem, but eventually contour lines became the conventional way of expressing change in elevation. In essence traditional pen-and-paper cartography evolved as a way of coding the features of the world in symbolic form.

Like many other areas of human activity, the various stages of map-making from data acquisition to compilation and editing and eventual distribution have become computerized, and now take advantage of many of the benefits of the digital world: transmission at electronic speed, automated numerical calculation, and easy copying and editing. People have learned how to express geographic knowledge in the binary alphabet required by digital computers, and have adopted a number of standard procedures that

[1] National Center for Geographic Information and Analysis, and Department of Geography, University of California, Santa Barbara, CA 93106-4060, USA. Phone +1 805 893 8049, FAX +1 805 893 3146, Email good@geog.ucsb.edu

allow individuals separated by large distances and by differences of culture and language to communicate that knowledge effectively.

Essential to these advances are the various rules used to express geographic knowledge in binary code – the equivalent for the geographic world of music's MP3, and the digital world's replacement for the conventions of pen and paper. Several distinct phases can be identified in the development of these digital representations, or *data models* to use the conventional technical term, each achieving a significant advance on the previous one by being able to capture and express a wider range of geographic phenomena with greater fidelity. However no representation can possibly be perfect, since the real world is infinitely complex. Over the past two decades much attention has been focused on the concept of uncertainty, or the degree to which a representation leaves its users uncertain about the true nature of the real world (Zhang and Goodchild, 2002). There are many sources of uncertainty, including measurement error, vagueness in the definitions of terms, and the need to force information into the template provided by a data model. Most advances in representation and in the technologies of data acquisition provide a closer approximation to the real world, but the ideal of perfect representation is necessarily unachievable.

This paper provides a brief history of the digital representation of geographic information, and the advances that have been made. While each advance is an improvement on its predecessors, removing constraints, expanding the set of geographic information types that can be represented, and reducing inherent uncertainty, the goal of a comprehensive representation of all geographic information remains distant. The next three sections describe the major phases, each following from the adoption and adaptation of improved data modeling concepts from the computing mainstream. This is followed by an assessment of the current state of the art, represented by the object-oriented paradigm, and of recent advances in data-modeling research. A cyclical model is presented, in which new adoptions are followed by increasing stress, as geographic information types for which the adopted approach is essentially inappropriate are nevertheless forced into its template, and finally by replacement with a new, more powerful approach. The paper ends with speculation about the prospects for a new cycle.

THREE PHASES

Flat files

One claimant to the title of first geographic information system (GIS) is the Canada Geographic Information System (CGIS), developed by the Government of Canada with the assistance of IBM in the 1960s. It was motivated by the Canada Land Inventory, a massive effort to map the capabilities and current use of land within a large swath of Canadian territory. The primary objective of the inventory was to provide statistics on the current and potential use of land, and the amounts of land available for new uses. This required a large mapping effort, followed by a detailed analysis based largely on the measurement of area.

Within this domain it was possible to argue that all maps were essentially of the same type: they showed land divided into irregularly shaped areas, each with homogeneous characteristics. For example, the maps of soil capability for agriculture divided geographic space into areas of uniform capability, measured on an ordinal scale and qualified by numerous codes. Even today, the majority of mapping of soils, surficial geology, land use, land cover, vegetation type, and habitat are of this type (Figure 1), which Mark and Csillag (1989) term the *area-class map*, though other terms such as *polygon coverage* are also in use. More formally, such maps can be regarded as depicting functions that map location $\mathbf{x}$ to one of a set of k classes, $c = c(\mathbf{x})$, in other words a nominal field.

[Figure 1 about here]

At the time, the only practical medium for large-volume digital storage was the magnetic tape, a linear structure that required substantial tape movement to access records in other than their stored order. The designers of CGIS faced a significant problem: how to represent the contents of an area-class map as a linear sequence of binary digits. This could clearly be done by scanning, if a suitable device could be developed that would convert the maps to raster representation, and if the cells of the raster could be represented row by row using some appropriate scheme to convert the brightness of the map in each cell to a set of binary digits. But the requirements of CGIS and the interests of efficient performance seemed to indicate a different approach that would focus on the boundary lines on the map. In the *vector* approach a line is represented as a series of points connected by straight segments, creating what is known as a *polyline*. A complete patch is represented by a connected sequence of such polylines, forming a polygon.

Thus one might consider representing an area-class map as a set of polygons, each composed of straight segments connecting points, the contents of each being described by a set of coded attributes. In the terminology of mainstream computing such a solution would create a *flat file*, a collection of records of the same type. Such records would be of variable length, because polygons would vary in shape complexity and thus in the numbers of points required to code them, but their representations could be laid end to end on magnetic tape.

In practice this solution turns out to be quite inconvenient, because it results in every internal boundary of the map being represented twice, resulting in unnecessary duplication. Moreover if the data are edited or transformed in any way, it is difficult to ensure that the two versions of each internal boundary remain identical. Instead, the designers of CGIS adopted a solution that is far from intuitive but results in much improved performance. In this solution the basic record is not the polygon, but the section of common boundary between two polygons. Each such record contains descriptions of the polygons on each side, along with the points required to describe the common boundary's geometry. Various terms have been assigned to these polylines, including *arcs*, *chains*, *edges*, and *links*, the first being responsible for half of the name of the most popular commercial GIS software, ArcInfo. In CGIS these records formed a flat file, and were recorded in sequence on magnetic tape. Many of the analyses needed by CGIS, such as measurement of polygon area, turn out to be more efficiently performed on arcs rather than polygons, even though the arc-based solution contains no direct representation of the latter.

This flat-file solution worked well for CGIS, which despite initial teething problems was by 1975 in full production (Foresman, 1998). Maps of current land use, soil capability, or capability for recreation could all be expressed in the same basic form as area-class maps. Systems that came later adopted the same basic approach, which could also be applied to maps of administrative areas, census reporting zones, or forest stands; each of these aspects of the geographic world could be represented as a collection of common boundaries between areas of homogeneous characteristics. It was possible to represent holes and islands, in other worlds polygons unconnected to the rest of the boundary network, if imaginary "causeways" could be inserted to connect them to the network, and ignored during analysis or visualization (such causeways are easily recognized because the same polygon appears on both sides). Road networks could be represented if one relaxed the requirement that every arc end in two junctions (Figure 2), though overpasses and underpasses presented an additional problem. Data sets with complex attributes, such as those that result from the reporting of census summaries, were also a problem since attributes had to be repeated for every arc surrounding a polygon. When attributes changed or were edited it was difficult to ensure that every copy remained identical.

[Figure 2 about here]

The arc-based solution adopted by CGIS was elegant, but it nevertheless did some damage to the truth as represented by the source map, which in turn simplified the real world. While the source map could show curved boundaries, these had to be rectified into polylines in the digital system. Moreover, the map's use of homogeneous areas to describe what were essentially continuous variations of phenomena over the Earth's surface added to the uncertainty inherent in the representation. Much effort has been expended in devising ways of representing this uncertainty. The *confusion matrix*, for example, records comparisons between the characteristics found at points on the Earth's surface with the characteristics recorded for those points in the data, and summarizes the table in simple statistics such as Kappa (Stehman, 1996).

The relational model

In the 1970s a new data model emerged in the computing mainstream that was to dominate thinking for the next decade. Tape storage was slowly giving way to disk, which allowed randomly selected records to be retrieved without the delays involved in winding tape. The relational model exploits this random-access ability, and is based on the following assumptions:

1. All information can be expressed in the form of tables or *relations*, each row defining one record, object, or case, and each column defining one of the characteristics of each record, object, or case. Clearly one can make such a table of the polygons of an area-class map, using one row for each polygon and one

column for each of the known attributes of the polygon. However polygons have a variable number of coordinates, so the table concept is not suitable for describing each polygon's geometric shape.

2. A given database can include many tables, recording the properties of different types of objects. Tables are linked through common keys or pointers; for example a record in a table of patients might be linked to (point to) a record in a table of doctors, or a record in a table of passengers might be linked to a record in a table of flights.
3. The database can be *normalized* through a series of formal steps designed to ensure that no information is duplicated unnecessarily. For example, in an airline reservation system the information that details flights is stored not with the passenger records but in a separate table of flight records. In this way changes to flight details can be made once, rather than by editing the records of every passenger taking the flight.

The design adopted by ESRI for its first version of ARC/INFO represented a very significant step beyond the flat files of CGIS, by exploiting the relational model. The INFO database management system, one of the early commercial products to implement the relational model, was used to create a much more powerful solution to the problem of representing area-class maps. Two linked tables were created in what became known as the coverage model (Figure 3):

1. A table of polygons, recording the attributes of each one as a series of columns.
2. A table of arcs, each one linked to two polygon records by common keys, one pointing to the polygon on the left of the arc and one to the polygon on the right (left and right are defined by the order in which the arc's sequence of points is stored). Arc records also included pointers to the nodes or junctions at each end of the arc, though these nodes did not have their own table in early versions.

[Figure 3 about here]

Because the structure stores explicit representations of the relationships between arcs, polygons, and nodes it is often described as *topologically rich*, or simply *topological*. But the coordinates needed to define each arc's geometry could not be stored in the structure, and instead were stored in a uniquely structured file outside the relational model. This solution, which was adopted by ESRI and is reflected in the choice of ARC/INFO as the name of the software, has been termed the *hybrid* model (DeMers, 1997) for this reason. Almost three decades later, far more powerful and flexible relational database management systems are able to accommodate a much wider range of data types in the cells of their tables, including complex representations of polylines and polygons, and the need for the hybrid solution has largely disappeared.

While this adaptation of the relational model, or *geo*-relational model, could clearly accommodate area-class maps, it almost immediately came under pressure from applications involving other geographic data types that did not fit the area-class model. As noted earlier, it is possible to fit road networks into the model provided that special allowances are made for cul-de-sacs, which end at otherwise unconnected nodes (nodes of valency 1). Moreover, the model cannot distinguish between overpasses, underpasses, and intersections at grade, since nodes must exist at every crossing. This latter problem was eventually addressed by creating a *turntable*, and explicitly enumerating all of the turns that could be made at any crossing. Transportation applications also require the ability to position landmarks, accidents, and other point-like events at points on arcs that may not be nodes. This was addressed by the concept of *dynamic segmentation*, which allowed features to be located using a system of *linear addressing* based on identifying the relevant arc and the distance along it from its starting node.

Several further extensions were made to the basic model. Data about points could be handled by allowing nodes that were not connected to the boundary network (nodes of valency 0), or by allowing polygons of zero area. Changes to a boundary network through time could be handled by creating a single database of all boundaries that ever existed, and adding a *region* data set that defined how polygons should be constructed from these primitive pieces at any specific date. In this way it was possible to accommodate the changing boundaries of the U.S. census, easements on property, overlapping wildfires, and many other complex phenomena.

In short, in the two decades following the adoption of the geo-relational model a series of extensions were made that accommodated new applications, but at the same time created an increasingly unmanageable superstructure. The software needed to handle the extensions became increasingly complex, terminology became increasingly confusing, and GIS software began to resemble a house of cards, with hundreds of basic concepts and thousands of commands.

By contrast, the approaches being used in computer-assisted design (CAD) software were far simpler and easier to understand. GIS software that adopted this simpler approach began to compete in the marketplace, putting pressure on the mainstream GIS software industry to find similarly simple approaches. When ArcView appeared in the mid 1980s it was marketed as an easy-to-learn entry to GIS, with the hope that its users would eventually migrate to the more powerful ARC/INFO. Its data-model template recognized features on the Earth’s surface as points, lines, or areas, respectively represented digitally as points, polylines, and polygons. No topological relationships between these features were accommodated, however, since the software was intended only for visualization of data, and no support was provided initially for the operations that benefit from topological structure: digitizing, editing, and analysis.

In time, however, pressure built to add such capabilities, and at the same time computer power and storage capacity grew, making it no longer imperative to avoid the double internal boundaries of area-class maps, and making it possible to compute topological relationships as and when required. ESRI also departed from past practice by publishing the format of its ArcView files (the *shapefile* format).

In summary, the early 1990s were characterized by a proliferation of data models. The original topologically rich coverage model had become increasingly compromised by numerous extensions, while the shapefile model had provided the user community with a comparatively simple alternative. Some vendors had attempted to replace this complexity with a simpler, more uniform solution, but none had achieved significant market share. Conditions were ripe for a new approach that would sweep away much or all of the complexity, and introduce a more coherent data model grounded in a better understanding of the nature of geographic data.

Object-oriented modeling

Mainstream thinking about data modeling had advanced a long way in the two decades following the popularization of the relational model. Object-oriented data modeling offers a more comprehensive approach that accommodates several concepts missing from the relational model, based on the following assumptions (Zeiler, 1999; this discussion has been adapted to the particular needs of GIS):

1. All things, cases, events, instances, or objects of interest can be placed into classes.
2. Every member of a class can be distinguished on the same set of characteristics or attributes.
3. Classes can be specializations of more general classes and *inherit* their properties.
4. The members of a class can be aggregations of members of other classes, or composed of members of other classes.
5. The members of a class can be related to members of other classes through associations.
6. Methods can be associated or *encapsulated* with classes.

For example, all of the 50 states of the U.S. are members of the class *state*. States have numerous distinguishing attributes, including area, name, population, and date of admission into the Union. States are polygons, and have all of the characteristics of polygons (area, perimeter length), and also have more specific properties that polygons in general do not have (population, name). States are composed of counties, and have many associations; for example, an association exists between U.S. cities and their containing states (every U.S. city lies in exactly one state, while a state may contain any number of cities including 0).

The object-oriented approach proved to be far more powerful and flexible than the geo-relational model. It introduced the hierarchical concepts of inheritance, aggregation, and composition that had been entirely absent in the earlier solution, and allowed GIS database designers to capture the essentially hierarchical nature of many geographic phenomena, from the administrative hierarchies of township–county–state–nation to the scale hierarchies of river and road networks. It provided a general framework for many of the problems that had previously been addressed through special extensions: regions, dynamic segmentation, and temporal change. It allowed topological relationships between objects to be handled much more flexibly, and it allowed editing rules and other constraints to be represented as methods encapsulated with classes.

ESRI introduced object-oriented modeling in 1999 in ArcInfo Version 8, and later merged ArcView with ArcInfo into a single product line. Just as with the introduction of the relational model from the computing mainstream in the early 1980s, object-oriented modeling provided a uniform solution to an accumulation of problems, and a fresh, more powerful start on the representation of geographic phenomena. The basic classes of GIS—polygons, polylines, points—were specialized in a series of major GIS application domains, allowing users in those domains to access a standard template that had been designed to accommodate all of the classes commonly encountered in that domain (Arctur and Zeiler, 2004). For example, the UNETRANS data model developed for transportation applications includes specializations of polygons to Traffic Analysis Zones and specializations of polylines to rail lines, bicycle paths, and canals. Object-oriented models have been constructed for applications that have never been associated with maps, helping to move GIS further away from its dependence on the map metaphor (Goodchild, 1988), and towards a comprehensive approach to the representation of all types of geographic information.

Remaining issues

Despite the success of object-oriented data modeling, there are reasons to believe that the story is not yet finished, and that a new round of innovation may be needed. Two arguments lead in this direction, both grounded in the fundamental realities of the geographic world. Both arise as objections to the first assumption of object orientation given above, with its implication that all geographic phenomena can be conceptualized as things, events, cases, or objects—in other words as discrete. This assumption is as fundamental as the nature of computing itself.

Nevertheless there are numerous phenomena on the Earth's surface that are fundamentally continuous, and for which discretization, or the breaking of phenomena into discrete pieces, is to some degree inappropriate or problematic. Roads, for example, are continuous, but are typically broken into pieces at their intersections and represented as collections of discrete polylines. Rivers are similarly segmented at junctions, or at real or imagined breaks, into reaches. More generally, it has long been recognized that humans approach the geographic world in two distinct ways (Longley *et al.*, 2005). In the first, the *discrete object view*, the metaphor of an empty tabletop is used to conceptualize the world as empty except where it is occupied by discrete, countable objects that may or may not overlap and may or may not cover the table. Biological organisms, vehicles, and buildings fit this model well. On the other hand, other phenomena are better conceptualized as *continuous fields*, in which every location $\mathbf{x}$ is mapped to a single property z that may be nominal, ordinal, interval, or ratio; and scalar, vector, or tensor. Properties such as elevation, ownership, wind speed and direction, soil class, and current land use fit this model well.

To handle representations of continuous fields in GIS it is first necessary to discretize their continuous variation, using one of a number of approaches. Six of these are commonly implemented in GIS, while many others are in common use in specific scientific domains. But the resulting discrete objects are indistinguishable from collections that represent phenomena conceptualized as discrete objects. The result is inconsistency, complexity in the user interface, and an environment in which it is easy to make errors. Longley *et al.* (2005) use the example of eight points, which might represent eight cities or eight weather stations, but are otherwise indistinguishable as GIS data sets. In the second case, which implies a continuous-field conceptualization, it is reasonable to apply methods of spatial interpolation to estimate atmospheric properties in the spaces between the stations. But in the first case, which implies a discrete-object conceptualization, it would clearly be absurd to interpolate a property such as population count.

In the world of object-oriented data modeling these issues should in principle be handled by methods encapsulated with each class. Thus if a set of polylines represents digitized elevation contours, a method should prevent any edit that will result in contours crossing. Similarly a method should be associated with a set of polygons representing states to prevent any edit resulting in an overlap or gap between adjacent states, since the property *state* is conceptualized as a nominal field with exactly one value at every point within the national boundary. Two types of polygons should be distinguished, one related to discrete objects and the other to continuous fields; and in the second case all specializations should inherit the appropriate methods. Similar strategies should be adopted with respect to polylines and other classes that can be used to discretize fields. To date, however, no such implementations have been described.

While the field/object distinction is powerful and covers a wide range of phenomena, there is now a recognition that other phenomena may not fall neatly into either category. Cova and Goodchild (2002) describe *object fields*, in which every location in space–time maps to an entire object, and show how viewsheds, watersheds, and trade areas are of this nature. Yuan (2001) has described *field objects,* or objects with continuously varying internal structure. Time adds a new dimension to this discussion, because of the many types of temporal change (Peuquet, 2002), as does the third spatial dimension.

The second problem concerns the earliest stages in the acquisition of geographic knowledge. Both relational and object-oriented approaches are based on tables, and imply the existence of well-defined sets of objects and attributes. But while this model may fit well to mature mapping processes, it leaves much to be desired as a representation of geographic exploration, and the processes by which human observers build conceptual understandings of the world around them. Classification is often regarded as the first stage of any scientific analysis, but there are certainly stages of observation that occur well before the establishment of classification schemes. And while maps identify *features* on the Earth's surface, the identification of features represents a fairly mature level of understanding of a landscape, and a high degree of consensus.

Consider, for example, the explorations of Lewis and Clark, or the fieldwork of Alexander von Humboldt or Charles Darwin. All of these explorers made extensive records of their travels and observations, but made little use of the tables that dominate in geographic data modeling. Instead, one might characterize their observations as largely unrelated and uncoordinated notes of the form $<\mathbf{x},z>$, indicating that at some location in space–time **x** a property *z* was observed. Only later was it possible to assemble these unrelated observations into tables, and to conduct the kinds of analysis now associated with GIS. Goodchild, Yuan, and Cova (2007) have termed this the *atomic* form of geographic information, and have shown how discrete objects and continuous fields can be conceptualized from many such atoms, along with object fields, field objects, and the tables of high-level geographic data modeling.

Conclusion

Flat files, relational models, and object-oriented models represent three stages in the evolution of geographic data modeling. Each approach is more comprehensive than the one it replaces, and while GIS developed initially based on the realization that several types of mapped data could be represented using the same basic approach, today's object-oriented data models encompass a much wider range of geographic data types that extend far beyond the traditional domain of cartography. Each approach has emerged from the mainstream computing industry, as computers became more powerful and as the technology of database management became more sophisticated; and each approach has been quickly adopted and adapted to the needs of geographic information.

No single approach can accommodate all data types, and the number of such types continues to increase as geographers and others explore the modeling of complex, dynamic phenomena. In each of the three stages pressure has built to accommodate a wider range, and work-arounds and extensions have been added, increasing the complexity of the approach and reducing its essential coherence. Finally a new solution has emerged from the mainstream, and has been adopted with enthusiasm, restarting the cycle.

The final section argued that the current emphasis on object-oriented designs is inadequate in two respects. It fails to model continuous fields in appropriate ways, and to prevent users from confusing data sets based in continuous-field and discrete-object conceptualizations. Moreover it fails to accommodate the earliest stages of field observation, leaving a large and important phase of scientific research without effective formalisms and computational support.

Whether the cycle will begin again remains to be seen. Previous cycles have been initiated by developments in the mainstream computing industry, and there are no indications that mature technologies are about to emerge to solve either of these problems. Moreover the GIS software industry is driven by its largest commercial customers, many of whom operate in areas such as asset management where discrete-object conceptualizations are more appropriate. Nevertheless the inability to handle continuous fields effectively is a cause of substantial confusion in GIS applications, and significantly increases the difficulties of learning GIS. Perhaps the use of encapsulated methods, as proposed in this paper, will provide a short-term solution.

The geographic world is infinitely complex, and its useful and accurate representation in the limited and discrete space of a digital computer remains a challenging problem, just as earlier the need to represent the world using pen and paper had its own severe limitations. The formal nature of a database inevitably favors certain types of geographic knowledge over others, and tends to work against knowledge that is subjective, inconsistent, and otherwise at variance with the principles of scientific measurement. Much progress has been made in the four decades since the first GIS experiments, but much important research and development remains.

ACKNOWLEDGMENT

The support of the U.S. National Science Foundation through Award BCS 0417131 is gratefully acknowledged.

REFERENCES

Arctur, D. and M. Zeiler, 2004. *Designing Geodatabases: Case Studies in GIS Data Modeling*. Redlands, CA: ESRI Press.

Cova, T.J. and M.F. Goodchild, 2002. Extending geographical representation to include fields of spatial objects. *International Journal of Geographical Information Science* 16(6): 509–532.

DeMers, M.N., 1997. *Fundamentals of Geographic Information Systems*. New York: Wiley.

Foresman, T.W., editor, 1998. *The History of Geographic Information Systems: Perspectives from the Pioneers*. Upper Saddle River, NJ: Prentice Hall.

Goodchild, M.F., 1988. Stepping over the line: technological constraints and the new cartography. *American Cartographer* 15: 311–319.

Goodchild, M.F., M. Yuan, and T.J. Cova, 2007. Towards a general theory of geographic representation in GIS. *International Journal of Geographical Information Science* 21(3): 239–260.

Longley, P.A., M.F. Goodchild, D.J. Maguire, and D.W. Rhind, 2005. *Geographic Information Systems and Science*. Second Edition. New York: Wiley.

Mark, D.M. and F. Csillag, 1989. The nature of boundaries on ‘area-class’ maps. *Cartographica* 26: 65–77.

Peuquet, D.J., 2002. *Representations of Space and Time*. New York: Guilford.

Stehman, S., 1996. Estimating the kappa coefficient and its variance under stratified random sampling. *Photogrammetric Engineering and Remote Sensing* 62: 401–407.

Yuan, M., 2001. Representing complex geographic phenomena with both object- and field-like properties. *Cartography and Geographic Information Science* 28: 83–96.

Zeiler, M., 1999. *Modeling Our World: The ESRI Guide to Geodatabase Design*. Redlands, CA: ESRI Press.

Zhang, J.-X. and M.F. Goodchild, 2002. *Uncertainty in Geographical Information*. New York: Taylor and Francis.

FIGURE CAPTIONS

1. An area-class map; part of the CalVeg map for the Santa Barbara area, showing areas of approximately homogeneous vegetation type.

2. Major roads of the Los Angeles basin. It was possible to handle dead ends by extending the basic model devised for area-class maps.

3. An example of a topological structure. The map contains four polygons and nine arcs. The tables display the contents that would be stored in a relational-model representation, with pointers from arcs to the polygons of which they are part.

Polygon table

ID	Name
A	Jasper
B	Newton
C	Pocahontas
D	Greenbrier

Arc table

ID	Left	Right
1	D	
2	B	D
3		B
4	A	
5	B	A
6	C	A
7		C
8	D	C
9	C	B