Title
Two Perspectives on Geographical Data Modelling (90-11)

Permalink
https://escholarship.org/uc/item/7zn585sw

Authors
Frank, Andrew U.
Goodchild, Michael F.

Publication Date
1990-11-01

NCGIA

National Center for Geographic Information and Analysis

Two Perspectives on Geographical Data Modelling

by

Andrew U. Frank and Michael F. Goodchild

National Center for Geographic Information & Analysis
University of Maine, Orono
University of California, Santa Barbara

National Center for Geographic Information & Analysis/NCGIA

Technical Paper 90-11

November, 1990

INTRODUCTION

Andrew U. Frank[1]

The two articles that comprise this technical report address a fundamental problem for geographic information, namely the modeling of space, from two different points of view. In today's GIS a number of methods to model space are used, often without sufficient theoretical analysis. In geographical research, appropriate concepts for modeling space are used, again often without a discussion of their implicit simplifications and restrictions. In his paper, Michael Goodchild addresses this problem by proposing a single unifying spatial concept to which many other concepts can be linked. Andrew Frank's article describes the building of a framework for the terms *spatial concepts*, *geometric data model* and *spatial data structure*, and defines theses notions and gives examples.

The two articles are similar in their description of different spatial conceptual methods to model space and different geometric data models used in GIS. Goodchild stresses the potential for these models to be linked to a concept of geographic space, seen as an infinite collection of points with a set of attributes related to each point. Discretization is the principal means for the standard geometric data structures (e.g. raster, networks etc) to be constructed. This conceptual unifying model is then useful to connect datasets expressed in different models, and it also leads the way to analytical treatment of error and error propagation. Frank is more concerned with the difference between conceptual models - like the geographic space Goodchild describes - and the geometric data models, which can be used for implementation of GIS. A geometric data model must be formally described and, in principle, must be implementable; therefore it is necessary a discrete structure. By analogy to the use of the term *data model* in the database management system literature, it is proposed to use *geometric data model* for the reference model and reserve the term *spatial data structure* for the actual implementation of a geometric data model. The data model describes the logical operations and defines their results, and the data structures show how these can be realized and address specific performance issues.

The two papers are closely related to the research agenda of the NCGIA and bring together the insight gained from different research efforts. Michael Goodchild leads research initiative 1, Accuracy of Spatial Databases, where a concern with modeling error and error propagation in spatial data led him to consider the spatial concept he now proposes (Goodchild 1989). Andrew Frank as a co-leader of initiative 2, Languages of Spatial Relations, focuses on the separation of conceptual and formal models for space and on the need for linkages between the different models used. The articles are related to initiative 3, Multiple Representations, where the differences between the various representations of the same geographic features are often found exactly in the spatial concepts used for modeling. In initiative 4, Use and Value of Geographic Information, a taxonomy of geographic information was deemed necessary.

Other work by researchers from the NCGIA related to these issues can be found in:

- a book edited by Goodchild and Gopal containing the papers presented at the Initiative 1 meeting (*Accuracy of Spatial Databases* Taylor and Francis, London);
- the report on the initiative 2 meeting (NCGIA Technical Reports 89-2 and 89-2A);
- the report on the initiative 3 meeting (NCGIA Technical Report 89-3);
- the report on the initiative 4 meeting (NCGIA Technical Report 89-6);
- the report on the initiative 5 meeting (NCGIA Technical Report 89-13);

The two articles in this report were originally written for a meeting organized by the Midlands Regional Research Laboratory at the University of Leicester from 21-22 March 1990 on the topic of GIS Design Modelsand Functionality. The meeting brought together a number of researchers and developers, from universities and corporations in the UK and the United States with mutual interests in the models underlying current and future GISs. It is hoped that many of the papers presented at the meeting will eventually be published in the journal *Computers and Geosciences*.

REFERENCES

Goodchild, M.F., 1989. Modeling error in objects and fields. In M.F. Goodchild and S. Gopal (eds.) *Accuracy of Spatial Databases*. Taylor and Francis, London: 107-14.

---

[1] National Center for Geographic Information and Analysis, University of Maine, Orono, ME 04469. (207) 581-2149. Email: frank@mecan1.bitnet

GEOGRAPHICAL DATA MODELING

Michael F. Goodchild[2]

Abstract

Data modeling is defined as the process of discretizing spatial variation, but is often confused with issues of data structure, and driven by available software rather than by a concern for accurate representation. We review the alternative data models commonly available in spatial databases, and assess them from the perspective of accurate representation of geographical reality. Extensions are discussed, particularly for three dimensions and time dependence.

INTRODUCTION

Tsichritzis and Lochovsky (1977 p.21) define a *data model* as 'a set of guidelines for the representation of the logical organization of the data in a data base...(consisting) of named logical units of data and the relationships between them.' With few if any exceptions, the world which is represented in a spatial database is not composed of logical units, and thus must be abstracted, generalized or approximated in the process of creating a database. Data modeling thus plays a fundamental role in spatial databases, and controls the view of the world which the user ultimately receives. As the GIS industry matures, and questions of data structures, algorithms and functionality become standardized, the critical issue of data modeling will become more and more important, both directly and indirectly through the role that it plays in such concerns as accuracy. Ultimately, a GIS will only be successful if it can present the user with an accurate view of the world; to do so requires both efficient access to a database, and the use of accurate data models. Moreover different forms of analysis and exploration of the same area will likely require different data models, depending on the form of approximation adopted in each.

The purpose of this paper is to examine the issue of spatial data modeling not from the perspective of alternative data structures, but as a process of representing geographical reality. We argue that the existence of alternative data models is one distinguishing feature of spatial databases, and creates the need for this distinct perspective. Too often the choices between data models are presented as choices between data *structures*, or specific arrangements of records and linkages within the database. If a data model consists of 'named logical units of data' and such logical units are abstractions or approximations of geographical reality, it follows that one data model is not necessarily obtainable from another, since each may approximate reality in different ways.

To illustrate the distinction being drawn here between data models and data structures, consider a simple raster in which each pixel has associated with it an integer representing a census tract number. These census tract numbers point to the rows of a table containing socioeconomic data. We refer to this loosely as a raster data structure. Now suppose that a raster/vector conversion algorithm extracts the boundaries of each tract as polygons composed of vertical and horizontal pixel edges, and that each polygon now points to the same census tract table. We refer to this loosely as a vector data structure. Although the structure has changed, the information it contains is the same, as we have merely rearranged the components of the data model, and not changed the manner in which the model approximates reality. Perception of spatial variation is an important criterion in the development of data models for maps, whereas the choice of data models for spatial databases is likely to be guided by very different objectives.

For the purposes of the paper we use the term *geographical reality* to refer to empirically verifiable facts about the real world. Those facts may not be certain; in practice, many of the relevant definitions include substantial uncertainty, as for example in the land use class 'urban'. A data model is a limited representation of reality, constrained by the finite, discrete nature of computing devices; the term *discretization* conveys much the same meaning as data model in this context. In many cases the relationship between reality and database is complicated by the interposition of a map or analog store with its own data model. Filtering then takes place both between reality and the analog store, and between the analog store and the database. The data models available to analog maps are much more limited, as they are constrained by the technology of paper and pen (Goodchild 1988b), so the double filtering which takes place makes it even more difficult to present the user with an accurate view of reality.

The paper is organized as follows. We first discuss the nature of geographical reality, and the subsequent sections review alternative data models. Extensions to three dimensions and time dependence are discussed, and the paper ends with some implications

[2] National Center for Geographic Information and Analysis, University of California, Santa Barbara, CA 93106, USA. (805) 893-8049. Email: good@voodoo.bitnet.

with regard to data accuracy and error modeling.

GEOGRAPHICAL REALITY

The fundamental element of geographical information is the tuple T = $<x,y,z_1,z_2,...,z_n>$, giving the values of a set of n *spatial variables* at the location (x,y). We allow the variables z to be of any data type: binary, nominal, ordinal, interval or ratio. For variables measured on discrete scales, note that it is always possible to transform to a space of binary variables, or to collapse many variables to one. We assume each z to be single-valued at any location, thus excluding over-folded geological structures, which must be treated as three-dimensional. We assume that the tuple is empirically verifiable, for example by visiting the location (x,y). Later in the paper we discuss extension to the tuple $<x,y,h,t,z_1,z_2,...,z_n>$ where h is the vertical dimension and t is time.

Since x and y are continuous, the number of tuples is infinite. Thus data modeling can be seen as the process of reducing the number of tuples required to represent reality to some finite set small enough to be accommodated within the constraints of a digital store. We refer to the infinite set of tuples as a *field*. x and y are also continuous in analog map stores, but here the problem of data modeling is to find effective ways of representing the variability of z_1 through z_n. For example, it is desirable to find ways of representing as many real variables as possible in a single mapped variable, through transformations $f(z_1,z_2,...,z_n)$.

The variation of z_1 through z_n may be discrete or continuous in x and y. For parameters such as topographic elevation the data model may assume continuous variation, i.e. the absence of cliffs. However geographic surfaces typically do not have derivatives (or tangents) which are everywhere well-defined.

Spatial autocorrelation (Cliff and Ord 1981) plays a key role in the task of discretizing spatial variation. We observe in general that the similarity between the variables in the tuples $T_1 = <x_1,y_1,z_{11},z_{12},...,z_{1n}>$ and $T_2 = <x_2,y_2,z_{21},z_{22},...,z_{2n}>$ increases as the locations converge. Two general strategies for discretization emerge from this observation. The *sampling* strategy exploits spatial autocorrelation by assuming that (x_1,y_1) and (x_2,y_2) must be more than a certain minimal distance apart before the associated tuples are substantially different. The *piecewise* strategy assumes that the plane can be partitioned into homogeneous, simply-connected *regions*, with variation within each region described by some simple function.

Many data models are based on discrete *objects* located in the plane, allowing spatial variation to be represented by a set of tuples $<i,a_1,a_2,...,a_m>$ where i is an object and a_1 through a_m are the object's attributes. Location is described by a set of tuples $<x,y,o_1,...,o_i,...>$, where o_i is a binary variable indicating the presence or absence of object i at location (x,y). The next section reviews a number of such object representations. However in most cases objects are generalizations or approximations of variation and poorly defined. For example a soil map shows the variation in soil type over an area by defining a set of non-overlapping, space-exhausting area objects. But the locations of the boundaries are not well-defined, and soil type is only approximately homogeneous within each area (Mark and Csillag 1989; Fisher 1989). Thus neither set of tuples may be empirically verifiable - we cannot confirm that location (x,y) is within a given object, or that all points within the object have the given attributes.

Other objects such as benchmarks and buildings may be comparatively well-defined. Consider the infinite field of tuples <x,y,o> where o is a binary variable, value 1 if (x,y) is in the State of California, 0 otherwise. In this case the object is better defined, but there are still locations and levels of accuracy at which it is impossible to determine o without ambiguity, for example along the coastline. In effect, every representation of geographical reality based on discrete objects is approximate to some degree.

MODELS OF FIELDS

In this section we examine the alternative data models which have been exploited in spatial databases. Most of the discussion concerns the representation of a single variable, but multivariate issues are included at several points.

Piecewise models

Piecewise models partition the plane into simply connected regions, with variation described by a simple mathematical function in each region. Each location is assigned to exactly one region. Furthermore there exists at least one path between any pair of locations within the same region which is itself wholly within the region. Regions may therefore contain other regions, but may not be disconnected into islands. Many GIS data models implement the concept of a complex object, and thus allow the user to create a super-region as a union of several simple regions.

A number of forms are assumed for the function describing variation within each region:

Constant. In the simplest case the value of the variable is constant within each region. The number of possible values of the variable is now finite, at most equal to the number of regions, and the model therefore places no restrictions on the variable's data type. In some cases the regions are defined by the variable itself, by locating boundaries in areas of particularly rapid change, allowing the model to approximate what is in reality continuous variation (Mark and Csillag 1989). In these cases a second step of discretization is necessary in order to represent the continuous curves of the boundaries in digital form, most often by selecting a finite set of points and connecting them by straight lines. This form of discretization is merely convenient as the object being represented has no existence in reality.

In other cases the boundaries will have been defined by some process which is independent of the variable itself. For example, much socioeconomic data is discretized by using *reporting zones* with boundaries which follow streets, rivers, railroads etc. In this case also the discretization of boundaries is almost always by means of points connected by straight lines, although the nature of the phenomenon being represented would often suggest better alternatives. A number of systems allow arcs of circles as well as straight line segments.

Constant piecewise approximations are commonly used to describe spatial variations in soils, land use, land cover and many other biophysical variables. The identical data model is used for much socioeconomic data, although homogeneity within zone is less often assumed in analysis of such data. In the biophysical case it is common for each variable to produce a unique discretization, but in the socioeconomic case the set of boundaries is usually common to many variables.

Consider the set of biophysical variables $z_1,...,z_n$, each with its own associated set of regions $\mathbf{R_i}$. After discretization of variable i, each region is assigned an attribute a_i**?**S_i measured on a discretized version of the scale of measurement of z_i, where S_i denotes the discrete domain of the ith attribute. At any point (x,y) we can identify the attributes $<a_1,...,a_n>$ assigned to the regions which the point occupies in each discretization. The concatenated attributes at (x,y) are an element of the Cartesian product $S_1 x S_2 x ... x S_n$, and the associated regions **P** are the familiar product of topological overlay of the sets of regions $\mathbf{R_1},\mathbf{R_2},...,\mathbf{R_n}$.

The regions **P** can be obtained in two clearly distinct ways. First, each variable i can be discretized, and the resulting regions overlaid. Alternatively we might attempt to discretize the multivariate space defined by $z_1,z_2,...,z_n$ directly. In the space defined by the variables $z_1,...,z_n$ the first case would result in a partitioning by hyperplanes perpendicular to the axes; in the second, there would be no constraints on the geometry of partitioning. The terms ITU (integrated terrain unit) and LCGU (largest common geographical unit) are associated primarily with the first approach. As such, the debate over whether to use ITUs or independent discretizations as the basis for multivariate spatial databases (Burrough 1986 p.4) is essentially an issue of data structure rather than data model, and its resolution depends on the comparative costs of data processing and storage rather than on the accurate representation of reality.

Linear. Now suppose that the variation within each region is described by a plane, or linear function $a_0+a_1x+a_2y$. We require the scale of measurement of z to be continuous. The TIN model (Burrough 1986) is a particularly simple form of linear piecewise approximation where all regions are triangles, and nodes are restricted to triangle vertices. The major reason for adopting triangles is that it is easy to ensure continuity of elevation across triangle edges.

Although there are many examples of spatial variables measured on continuous scales, the TIN model is commonly applied only in the case of topographical elevation. Many landscape-forming processes are responsive to gradient, and tend to produce terrain with substantial areas in which gradient is constant, although glacial processes are a notable exception (Mark 1979). The TIN model allows the sizes and positions of triangles to adapt to the complexity of the terrain, with smaller triangles in rugged areas. Recent versions of TIN models allow the user to define TIN vertices interactively at critical points on the surface, and to position triangle edges along lines of observed discontinuity of gradient (McCullagh 1988).

Higher order functions. More complex functions offer the possibility of more accurate representation of variation within each region. Akima (1978) has described the use of quintic polynomials within each triangle of a TIN, with the advantage that gradients can be made continuous across triangle edges. However, although this may be useful as a means of producing visually acceptable contours from irregularly spaced point data, it is less so as an accurate depiction of terrain. Discontinuities of slope are common on real topography, and can be modeled with planar TINs by aligning triangle edges with observed ridges and valleys.

Contours

If the scale of measurement of z is continuous, and if the variable is strongly spatially autocorrelated, then the set of points $<x,y|z?c_1,c_2,...,c_m>$ defines a set of *contour lines*. Conventionally we assume that the contoured values $c_1,...,c_m$ are evenly spaced along the domain of z. The set of points forming each contour line can be further discretized as a finite set connected by straight lines. The resulting contour lines partition the plane into regions, in each of which the value lies between two consecutive contoured values. However the regions produced in this way have certain distinct characteristics imposed by the continuous nature of z. The associated boundary network has no nodes of valency greater than 2, and two regions can be adjacent only if the associated contour intervals share a common value.

The popularity of the contour model as a means of depicting topography on analog maps suggests that it is optimal for this particular variable and the technology of cartography. It is undoubtedly an efficient way of communicating information on the spatial variation of a continuous variable to the user, and is reasonably successful as a means of visualizing two variables. However as a means of discretization in the relatively unconstrained environment of spatial databases it has a distinct disadvantage, as the level of approximation varies dramatically across the plane, being maximum on contour lines and minimum midway between them. The accuracy at any point bears no relationship to the phenomenon, being controlled entirely by the arbitrary choice of contoured values. Nevertheless digitized contours continue to be a readily available source of topographic information, which is perhaps the most convincing example of the filtering effects of analog map data models on spatial databases.

Sampling

Spatial variation can be discretized by capturing the value of a variable at a finite set of points. A *raster* results if the points are uniformly spaced, while irregular sampling may be more efficient if the density of sampling can adapt to the local degree of variability. The TIN model provides a simple way of interpolating between irregularly spaced sample points in the case of a continuous scale of measurement. In general, however, sampling imposes no constraints on scales of measurement or on the range of values possible at each point.

PLANAR ENFORCEMENT

Thus far we have been concerned with the modeling of one or more variables whose values are defined everywhere in the plane. In many cases it is more convenient to view reality as an empty plane littered by objects, which may be points, lines or areas. Any location (x,y) is either empty or occupied by one or more objects. Each object has a set of associated attributes which serve to differentiate it from other objects. We have already commented that the quality of definition of such geographical objects is highly variable, and almost always less than perfect.

We use the term *planar enforcement* to refer to the rules used in converting this form of representation to a single-valued function defined everywhere. Planar enforcement occurs at many points in spatial data handling, and we consider three in particular, with associated examples.

Consider the task of digitizing region boundaries from a map. This operation creates a set of line objects littering the plane, or ‘spaghetti’. In order to create a set of regions in which every location has a single value it is necessary to first ‘snap’ lines at junctions and remove overshoots, and then to obtain the attributes of the regions thus formed in some consistent way. In some cases this latter step is achieved by assigning the attributes of an arbitrarily located point to the containing region, and in other cases by assigning attributes attached to each side of each line object. The GIS industry often refers to the whole operation as ‘building topology’, and the term ‘cartographic’ is often associated with the view of the world as a plane littered by (unrelated) objects.

Spatial interpolation is the term commonly given to the task of computing a complete, continuous surface from a set of sample points (Lam 1983). In this case a value is obtained everywhere in the plane from attributes attached to a finite set of point objects. Spatial interpolation can be defined for any scale of measurement, but is most often applied when the scale is continuous.

Now consider a set of objects lying in the plane with the attribute ‘woodlot’. We can safely assume that these objects will not overlap, but if they do there is no particular problem in assigning the same attribute to their intersection. But suppose the objects represent forest fires and the attribute is date. Because a location may have burned more than once it is not immediately obvious how the tuple <x,y,z> might be defined at each point. One alternative would be to make z a count of fires; another would make z the date of most recent fire, with a special code for unburned regions.

VARIATION ON NETWORKS

Thus far we have been concerned with the representation of variation over the plane, whether viewed as an infinite field of tuples or as a space littered with discrete objects. However neither of these views is particularly consistent with GIS applications in transportation or surface hydrology. In these areas data modeling requires two largely independent stages. The first models the network as a collection of objects embedded in the two-dimensional plane. The objects are typically nodes and links. In many applications it is important to distinguish between links which cross geometrically and links which intersect at a node, to allow for grade separations. In this sense planar enforcement may be inappropriate for these networks.

The second stage models the variation of phenomena along the network. Attributes may be associated with points, such as bridges or houses, or variation may be modeled by piecewise discretization. For example the variation of pavement quality over a highway network might be modeled by defining segments with homogeneous quality. Variation in elevation of railroad networks is commonly modeled by defining segments of constant gradient. We use the term *segment* to refer to a discrete element of a network.

In the first stage of discretization, locations are defined in the (x,y) plane. In the second stage, however, locations are more conveniently defined by a pair of the form <link,offset> such as street address, or the <route,milepost> addressing system used by railroads. In summary, the first stage requires the definition of a set of line objects; a location (x,y) may or may not be occupied by one or more objects. In the second, an infinite set of tuples $<l,o,z_1,z_2,...,z_n>$ is defined over the network, where l defines a link and o an offset distance from the origin of the link.

Just as the plane allows independent discretizations for each variable, it should be possible to define independent segmentations of a network without repeating the first stage of discretization in every case. Unfortunately many current GIS products do not allow this. Instead, both levels of discretization must be collapsed to one. Because link objects are allowed only homogeneous attributes in these systems, it is necessary to create nodes wherever a change of attributes occurs, in effect forcing the equivalent of an ITU strategy. For example a node must be positioned at every point event or change of attributes on a rail network, including stations, switches, tunnels, bridges etc., leading to almost endless proliferation of link objects.

In essence, transportation networks are not sets of linear objects littering the plane, but one-dimensional addressing systems embedded in two-dimensional space. The values of spatial variables are defined only on the network, and not in the intervening spaces. Similar issues of multiple levels of discretization exist in three-dimensional spaces and in the time-dependent case (see below). It is also possible to find examples of spatial variables whose values are defined only at points.

CLASSES OF OBJECTS

We have seen how spatial databases reflect two different views of reality - as infinite sets of tuples approximated by regions and segments (the field view), and as planes littered with independent objects (the object view). The concept of an object arises in both cases, but in the first the area and line objects representing regions and segments cannot exist independently, but instead must partition the plane and the network respectively, and must be organized into well-defined, planar-enforced layers. The rules affecting the behavior of objects in the two views are therefore different.

The term ‘object-oriented’ (OO) has received attention recently in the GIS literature (Egenhofer and Frank 1988a,b) as many of the computer science concepts of object-oriented programming and databases have stimulated discussion in a spatial context. The OO notion of object identity is clearly more compatible with the object view of reality than with the field view, and the systems currently being marketed as ‘object-oriented’ rather than ‘layered’ seem to be aimed at those applications in which the object view is more acceptable. Kjerne and Dueker (1988) have discussed the implications of the OO concept of inheritance in survey data, while Armstrong and Densham (1989) discuss the implications of the OO concept of encapsulation for spatial analysis and modeling. It will be some time before the full impact of OO on GIS becomes clear.

In summary, we define five classes of objects in two groups:

Field view: region, segment

Object view: point, line, area

The regions and segments in the field view must be grouped to collectively partition the plane or network, but the classification of objects in the object view is more flexible.

The relationships between objects fall into three types:

1. those which are necessary for the definition of objects, e.g. the relationships between points which define a line;
2. those which are computable from the geometry of objects, e.g. the ‘contains’ relationship between point and area, or the ‘crosses’ relationship between two lines;
3. those which are not computable, e.g. the ‘intersects’ relationship between two roads.

Finally, we introduce the concept of an *object pair* (Goodchild 1988a), a virtual object created from the relationship between two simple objects, which may itself have attributes. More formally, an object pair is the tuple $<i,j,a_1,a_2,...,a_m>$ where i and j are two objects of the same or different classes and a_1 through a_m are attributes. The attributes of an object pair are not normally associated with any simple object. For example, the relationship between a point object ‘sink’ and another point object ‘spring’ can have attributes of distance, flow, flow-through time, etc., but has no physical existence as a defined spatial object. Object pairs are important for modeling various forms of spatial information, and are implemented in ESRI’s ARC/INFO, for example, as the ‘turntable’ in the NETWORK module.

ACCURACY

General strategies

If the purpose of a spatial data model is to represent an infinite number of real tuples using a finite number, then accuracy is clearly an important issue in choosing between alternatives. Unfortunately the appropriate objective function to use in defining accuracy depends on the use to which the database will be put. It is tempting, for example, to assume that the appropriate measure for topographic elevation is the mean absolute difference between real and modeled height at a randomly chosen point, but this measure is much less useful than accuracy of aspect to someone concerned with modeling surface flow directions, which depend only on aspect.

In many cases there are several stages of discretization between reality and the database. For example we might model elevation by measuring spot heights photogrammetrically, drawing smooth contours, digitizing them and finally building a TIN model from the digitized contours. Information is lost at each step, and in some steps spurious information is introduced, particularly in the interpolation of smooth contours and the construction of the TIN.

Two general strategies seem appropriate. First, it is desirable to minimize the loss of information between reality and the database, by minimizing the number of discretizations and manipulations. In the example the data is probably already sufficiently discretized as spot heights, and the use of contours could be avoided altogether. Second, it is desirable as far as possible to serve diverse needs by creating several distinct views of the database. For example, the contour map could be derived as a view of the (spot height) database, rather than as a step in the complex input process. The ability to create customized views for different purposes should be one of the major advantages of a database approach. In principle, such cartographic devices as scale and generalization, which reflect different views rather than proper ties of reality, should be treated as far as possible as attributes of a given view of the database, rather than as attributes of the database itself.

Objects vs. fields

Although field and object views of reality are to some extent alternative perspectives, we have argued in this paper that the tuples of a field are empirically verifiable, whereas it is only possible to confirm the existence of an object imperfectly. For this and related reasons Goodchild (1989) has argued that error is more appropriately treated from the perspective of field rather than object. The objects compiled by cartographic processes are commonly stripped of any information on which a useful model of error might be based, by processes such as low-pass filtering of boundaries, and deletion of small regions.

The distinction between fields and objects in modeling error is best illustrated with the example of contours. We regard a given set of contours as a single sample from a population of possible sets, and require that there be no difference between two sets other than the effects of error. It would be unacceptable, for example, if one set were substantially longer or more wiggly than the other. Then it appears to be impossible to write down a stochastic process which would take one such sample and produce another by any form of distortion. For example, adding a Gaussian error in x and y to a randomly chosen set of points along each contour would clearly lengthen the contours, so the two versions would differ significantly. On the other hand, it is comparatively simple to take the field of elevations from which the contour objects were created, add a suitably spatially autocorrelated error field, and obtain a second set of contour objects.

THE RASTER/VECTOR DEBATE

The discussion to this point has deliberately avoided the terms ‘raster’ and ‘vector’, despite the fact that these are often used to characterize two distinct classes of GIS software. The first section of the paper argued that the same information could be readily restructured from a form which would be broadly labeled as ‘raster’ to one which would be accepted as ‘vector’. This section reexamines the raster/vector debate within the framework of data modeling established in the previous sections.

‘Raster’ refers to a data model based on a regular (usually rectangular) tesselation of the plane, in which all locational information can be imputed from a record’s sequential position, and is therefore missing from the data structure. However the geometry of a regular tesselation can be used to model spatial variation in numerous ways. First, and perhaps most commonly, the value attached to a pixel is assumed to apply homogeneously to the pixel’s entire area. In this case the raster model is a special case of the piecewise constant models described above. In the case of a DEM the value is most often an estimate of mean elevation within the pixel, and this is also the case with remotely sensed images. In other cases the pixel’s value is the modal or most common class, or the class at the central point. Further ambiguity arises in the case of a DEM, since local estimates of slope and aspect are often made by fitting a plane to a small neighborhood, implying a piecewise linear rather than piecewise constant model.

The term ‘vector’ is similarly ambiguous from a data modeling perspective. It can mean an irregular polygonal tesselation, with piecewise constant variation, or a TIN, or a set of contours, or an unstructured CAD file containing points, lines and areas. Thus neither ‘vector’ nor ‘raster’ provide unambiguous information on how the data models reality. A mapping exists between these terms and the alternatives discussed in this paper, but it depends on conventional usage and is very loose. Thus the assumption made in this paper is that ‘raster’ and ‘vector’ do not provide an adequate and sound basis for a discussion of data modeling.

EXTENSIONS TO THE MODEL

Complex objects

Several contemporary GIS designs include the possibility of defining complex objects as collections of simple objects. A complex object may have its own graphic transform - for example, a collection of points may be represented as a point - and its own attributes, some of which may be aggregations or means of simple object attributes. The mapping between simple and complex objects may be n:1 or n:m. The concept of a complex object is clearly incompatible with the field view of reality.

Shared primitives

In this extension an object such as a point or line may be shared between a number of objects. For example the common boundary between two polygons may follow a road: the road and the two segments of common boundary would be defined as a primitive shared between three objects. Any update of the shared primitive would thus modify all three objects. Again the role of this extension is different in the two views of reality. In the field view all arcs are by definition shared, and it would be inefficient not to treat them as common primitives. On the other hand sharing would not always be appropriate in the object view, even though the relevant objects might be coincident.

Inheritance and lineage

The model contains no explicit means for attributing accuracy or lineage to objects, or of propagating these attributes through GIS processes. The technology to do so is at this point very limited.

THREE DIMENSIONS AND TIME

The concepts of field and object views apply equally to three dimensional data (Ganter 1989), where one can visualize space as occupied by a collection of objects, or by a single-valued function. The B-rep and SOE options, which correspond roughly to vector and raster respectively, are alternative ways of structuring information, rather than alternative models. Any point in the space may be associated with the values of one or more functions in the tuple $<x,y,z,z_1,z_2,...,z_n>$, or by the presence or absence of one or more point, line or area objects. Alternatively variation may be defined over a surface or a set of lines embedded in the space.

Although the modeling of three-dimensional data presents no significant problems, and is common in such fields as CAD and medical imaging, it differs from the two-dimensional case in the lack of analogs. There is no equivalent of the map to act as an input medium or filter. Consequently, although 3D data models exist in the digital domain, the creation of data for them presents much more of a problem, and there is a general lack of suitable data for 3D GIS. High priority should be given to the development of a 3D data compilation workstation, which would allow a user such as a geologist to input evidence of various forms (well logs, seismic data, expert knowledge) and build 3D data models through a variety of forms of spatial interpolation. The same issue is missing in 2D because it is so easy for data compilation to take place using the map analog.

In the case of time, the asymmetry between time and the spatial dimensions introduces further complexity. There appear to be five major modeling options:

1. A finite number of discrete time slices, each viewed as a field.
2. Discrete time slices, with objects identified in each slice but with no information relating objects between slices.
3. Discrete time slices, with static objects which are present or absent in each slice.
4. Discrete time slices, with objects identified in each slice and linkages between corresponding objects at different times.
5. Continuous time, creating a three dimensional space in which the movement of two-dimensional objects is represented by three-dimensional objects (points become lines, lines become surfaces, areas become volumes).

DISCUSSION

In this paper we have tried to distinguish clearly between data structures and data models, defining the latter as alternative discretizations of the infinite complexity of spatial information. Data modeling in spatial databases appears to adopt two alternative views, depending on whether it regards reality as occupied by a set of single-valued functions defined everywhere, or by a set of objects. Although the data structures used in the two cases are often identical, they nevertheless represent very different perspectives on reality.

The choice of data model is critical in spatial data handling, because ultimately it affects the views which the database presents to the user, and which the user judges against empirical truth. Unfortunately, choices are too often limited by the set of models associated with analog maps, or by the set offered by the vendor. An effective choice between alternatives also requires a degree of understanding of the nature of the geographical phenomenon being represented, and the processes which created it. Finally, choice also affects the extent to which it is possible to model and understand uncertainty, and its propagation through the steps of analysis. Many of the supposed benefits of GIS technology - the ability to change scale and overlay, and the separation of the roles of data collection and analysis - are in some ways its greatest weaknesses.

REFERENCES

Akima, H., 1978. A method of bivariate interpolation and smooth surface fitting for irregularly distributed data points. *ACM Transactions on Mathematical Software* 4:148-59.

Armstrong, M.P. and P.J. Densham, 1989. Object oriented locational analysis. *Proceedings, GIS/LIS ’89*. ASPRS/ACSM, Bethesda, MD, pp. 717-726.

Burrough, P.A., 1986. *Principles of Geographical Information Systems for Land Resources Assessment*. Clarendon, Oxford.

Cliff, A.D. and J.K. Ord, 1981. *Spatial Processes: Models and Applications*. Pion, London.

Egenhofer, M.J. and A.U. Frank, 1988a. Object-oriented databases: database requirements for GIS. *Proceedings, International Geographic Information Systems Conference: The Research Agenda*. NASA, Washington, II:189-211.

Egenhofer, M.J. and A.U. Frank, 1988b. Designing object-oriented query languages for GIS: human interface aspects. *Proceedings, Third International Symposium on Spatial Data Handling*, pp. 79-96.

Fisher, P.F., 1989. Knowledge-based approaches to determining and correcting areas of unreliability in geographic databases. In M.F. Goodchild and S. Gopal, *Accuracy of Spatial Databases*. Taylor and Francis, London, pp. 45-54.

Ganter, J.H., 1989. A comparison of representations for complex earth volumes. *Proceedings, AutoCarto 9*. ASPRS/ACSM, Falls Church, VA.

Goodchild, M.F., 1988a. Towards an enumeration and classification of GIS functions. *Proceedings, International Geographic Information Systems Conference: The Research Agenda*. NASA, Washington, II:67-77.

Goodchild, M.F., 1988b. Stepping over the line: technological constraints and the new cartography. *The American Cartographer* 15:311-20.

Goodchild, M.F., 1989. Modeling error in objects and fields. In M.F. Goodchild and S. Gopal, editors, *Accuracy of Spatial Databases*. Taylor and Francis, London, pp. 107-14.

Kjerne, D. and K.J. Dueker, 1988. Modeling cadastral spatial relationships using Smalltalk-80. *Proceedings, GIS/LIS '88*. ASPRS/ACSM, Falls Church, VA, pp. 373-85.

Lam, N.S.-N., 1983. Spatial interpolation methods: a review. *The American Cartographer* 10:129-49.

Mark, D.M., 1979. Phenomenon-based data structuring and digital terrain modeling. *GeoProcessing* 1:27-36.

Mark, D.M. and F. Csillag, 1989. The nature of boundaries on 'area-class' maps. *Cartographica* 26:65-78.

McCullagh, M.J., 1988. Terrain and surface modelling systems: theory and practice. *Photogrammetric Record* 12:747-79.

Tsichritzis, T.C. and F.H. Lochovsky, 1977. *Data Base Management Systems*. Academic Press, New York.

SPATIAL CONCEPTS, GEOMETRIC DATA MODELS
AND DATA STRUCTURES

Andrew U. Frank [3]

Abstract

There seems to be some uncertainty in the GIS literature regarding the use of the terms *data model* and *data structure*. There is a clear understanding of these notions in the database literature and it is possible to define analogous terms for GIS: *geometric data model* and *geometric data structure*. *Geometric data model* is used to describe a formalized abstract set of spatial object classes and the operations performed on them. *Geometric data structure* is then the specific implementation of a geometric data model, which fixes the storage structure, utilization and performance. Humans organize their spatial perceptions using concepts that can be defined as spatial concepts to denote an informal or not directly implementable conceptual structure used to understand space. Examples are given to clarify the theoretical discussion.

INTRODUCTION

Discussions of data structures to model geometry for geographic information systems (GIS) have progressed considerably over the last 15 years. The key issue is to model geometric concepts describing reality using a computer system. Although this does not seem difficult, research and development efforts of recent years have often contributed more to an appreciation of the problem than to a final solution. Initially, the problem was considered one of optimal data structures on a very low level, close to the organization and operations of the basic computer hardware. Discussion of this topic can be found in Dutton (1979). Research during this time was concerned with the computer aided treatment of cartographic data and the industry produced computer assisted map maintenance systems. At the same time, there were papers discussing the analytical capabilities that a geographic information system could offer to geography and other geosciences. These functions appeared to be extremely attractive, but research indicated that models had to contain more than just the cartographic data.

Data structures to represent geometric data were also needed in CAD/CAM (computer aided design/computer aided manufacturing) systems. These systems were initially developed to facilitate the production of paper drawings (CAD) but with the promise of extending further into the design and manufacturing process. As in geographic information systems, the limitations of representing geometric concepts with the tools of traditional drawings became apparent.

Understanding the limitations of computer assisted map maintenance systems pointed the way to data structures which represent geometry, not the map image of geometric phenomena. Frank (1984) argued that there should be a differentiation between systems that deal with data directly representing some geometric reality and systems that deal with map representations. Only the former can support sophisticated geometric analytical functions, whereas the latter help human users to produce maps that can be analyzed by skilled users.

The discussion of geometric data structures often included treatments of the conceptual bases and the theoretical foundations but then detailed the implementation. For geographic information systems, two principal standard structures were established: vector and raster methods. Peuquet (1983) even proposed a compromise (vaster) concept. A very extensive literature for efficient implementation of raster structures using a quadtree data structure has been presented by Samet (1989a,b).

Efforts to establish a theoretical base for geometric data structures came from different quarters. A landmark work (Corbett 1979) stressed the importance of topology as a basic mathematical concept for organizing geometric data. This paper, unfortunately not published in a widely circulating journal, is otherwise typical for its time: it contains extensive discussion of implementation at the hardware/assembly language level, which somewhat obscures its deep theoretical contribution. Frank (1983a) found a graph theory based approach lacking. Peuquet (1988) used image processing concepts, and Chan and White (1987) traced the origin of the *map algebra* concept back to traditional methods used by urban planners.

In this context a number of issues relating to terminology arise. In the past, these issues have been the cause of some confusion and an attempt to resolve them is made here. It is noteworthy that geographers have started to read the database literature, where very similar problems have been dealt with for quite a number of years and terminology is well

---

[3] National Center for Geographic Information and Analysis, University of Maine, Orono, ME 04469. (207) 581-2149. Email: frank@mecan1.bitnet

established. Geographic information system should not invent new terminology, but use and extend by analogy, established information system and database terminology.

This paper will concentrate on the three notions of spatial concepts, geometric data models and geometric data structures. It will be shown that these are three different concepts which need to be separated. Each of the topics will be described in turn and some examples will be presented. The discussion will conclude with an overview of alternative viewpoints and the problems that can be resolved adopting the viewpoint presented here.

DATA MODELS AND DATA STRUCTURES

One of the reasons for building generalized database management systems was the observation that it was possible to program the low level data structures and the related access mechanism only once and make these generally useful methods available to many different applications. Work started with concepts like index sequential access methods (ISAM) and general purpose sorting and merging routines and progressed to hashing and tree structures. A complete and authoritative survey of all these data structures is given by Knuth (1973) for most ordinary (ie non-spatial) problems.

At the level of organization of data, early database management systems can be seen as generalized packages permitting the use of sort and search methods in an integrated package. Anyone who has tried to use a package of subroutines - and code, for these same functions are readily available today as packages of reusable routines - is well aware that the adaptation of such routines to a specific task is no minor feat. In order to describe the functionality of the database management system without including all details of the data structure etc, a simplified model of the data storage system was created. Most of the details of the specific data structure are implicit in this model. Indeed it was explicitly demanded that the data model should be generic and independent from the implementation or the specific hardware configuration in order to increase portability of an application and to ensure hardware independence of the application programs (Codd 1982).

Much of the early database management system discussion centers around the selection of the appropriate abstraction and data models, with the clear understanding that there is a trade off between higher level of abstraction and more automatic solutions vs lower levels of abstraction, more adaptability and thus (most often) higher performance. Different companies offered database management systems with different interfaces, with very significant differences in the ease of use or level of knowledge necessary to understand and use the system (CODASYL 1971).

Data models thus evolved from an effort to find the common functionality and provide an abstract model of typical implementations. In 1970 E.F. Codd defined a data model from a top-down perspective. He defined a conceptually simple data structure with an appropriate set of operations: the relational data model (Codd 1970). The stress on a data model thus focussed on a conceptually simple construction - which can be implemented in more than one way - and which explains the database management system behavior. From a database administrator’s point of view, the data model defines the interface from the database management system. It can thus be said, that ‘the data model defines the tools available to structure the data’ (Zehnder 1981) which will be stored in the database. This is essentially the same as the definition of the data model as ‘a set of guidelines for the representation of the logical organization of the data in a database’ (Tsichritzis and Lochovsky 1977 p.21).

From a modern point of view, it should be stressed that a data model is a set of objects with the appropriate operations and integrity rules formally defined (Ullman 1988; Date 1986). This is essentially the definition for an algebra and it is therefore appropriate to speak of a relational algebra. The specific object types are selected such that they can be used to explain or define the structure in data, and there is often a specific data description language defined. The concepts are selected so that they can be implemented.

The database community uses the notions of a data structure, which is a generic or specific set of methods or programs to access data stored in a specific way, and data model, which is a generic, highly abstract set of concepts with which a database administrator can describe the data and their relationships. We propose to use the same concepts for geographic information systems, arguing that geographic information systems face essentially the same problems and are constructed similarly to other information systems.

A data model is implemented by selecting a data structure which provides the operations defined for the data model and then establishing a mapping from the conceptual operations of the data model to specific programs that carry out the required operations.

DATA MODELS IN SPATIAL DATA: TWO EXAMPLES

In geographic information system research there have been numerous discussions of data structures which could be used to represent spatial data and provide a useful set of operations; in a recent set of books (Samet 1989a,b) a large number of such structures are surveyed. There is also a need for more abstract concepts to describe geometric data and the appropriate operations, which are independent from a specific implementation (Goodchild 1990; Frank and Mark 1991).

To clarify the notions of data model and data structure as applied to spatial problems, two major examples will be presented, namely the so-called raster and topological (vector) data models and their underlying data structures.

Raster data model

This popular data model is based on a raster which divides space in regularly shaped and sized pieces. For each of these pieces one then records attribute values, either as averages or the values at some specific points. There are a (small) number of variants in the raster data model, as we may use any of a number of regular tessellations to subdivide space (Diaz and Bell 1986). Typical operations on the raster data model combine the data from one raster cell (using the values for different properties) to compute a new data value for the same cell. This is a form of spatial overlay, which compares well with the practice of planners (Chan and White 1987).

There are a multitude of data structures which implement the raster data model, from the obvious use of a FORTRAN array, through run length encoding and quadtrees - as surveyed by Samet (1989a). The use of a regular square raster to represent geometric values is a geometric data model, for which we can define an appropriate set of operations independent of the specifics of an implementation - as was first done in map algebra by Dana Tomlin (1983, 1989). There are several methods to implement this geometric data model with its operations; quadtrees, with their specific variants of implementation, being among the most effective ones (Samet 1988).

Topological data model

Another frequently used data model is based on a subdivision of space into irregularly shaped regions (often called cells) with their boundaries, formed by lines called arcs or segments which link points (called nodes). This model is based on mathematical topology (Alexandroff 1961) and includes operations to find the boundary of a given object etc. For geographic information systems use one needs further an operation to overlay one partition with another one and to determine the intersection areas. Such an operation obviously uses metric properties to calculate the points of intersection between boundaries etc. Thus, the data model is not purely topological.

A standard implementation uses records for nodes (with their position expressed as coordinate pairs), records for areas with their values for the interesting properties, and records for arcs, which contain links to the start and end nodes for each arc and links to the area to the left and the right of the arc, as shown below. There are other implementation concepts that provide the same functionality, e.g. TIGRIS (Herring 1990) or the geo-relational algebra (Gueting 1988).

Nodes: <node-id, x, y>

Areas: <area-id, property-value1, property-value2,...>
Arcs: <arc-id, id of start-node, id of end-node, id of left-area, id of right-area>

In principle, results from operations on the same data, represented by different implemented data structures, should be the same. There were thus complaints when a federal agency tested geographic information systems which implemented more than one topological data model and found that the results from operations yielded substantially different results.

SPATIAL CONCEPTS

The data models of the database management system discussion were useful in representing a specific perception of the world, as represented by the data sets. This was often assumed as given, as these data represented artifacts (e.g. bank accounts, insurance policies, stock in warehouse) which were defined in an operative manner through the business practice. When software started to model real systems, i.e. a system which had an observable real counterpart, software engineers realized that there was an additional problem of how humans conceptualize reality. This is not a problem in most administrative applications as the business practice, rules and regulations define how things ought to be understood. But the problem is especially important with geographic information systems as humans seem to use several different methods to conceptualize space (Mark, Frank et al. 1989): we seem to use an essentially Euclidian geometry when we reason about the spatial arrangements on our table or other small areas, but use a network-topology view when we plan a trip or navigate a car, etc. It is not that reality changes but the concepts utilized to structure our perception of the situation may vary (Neisser 1976; Lakoff 1987). In order to cope with the complexity of a real situation we have to abstract from details and concentrate on the aspects that are important for the task at hand (Mark 1989; Frank and Mark 1990).

The concepts used to understand space are often based on notions which cannot directly be implemented, either for lack of formal definition or for lack of discretization. The imaging schemata (Lakoff and Johnson 1980), which are basic for spatial cognition, and include such fundamental spatial relations as inside, across etc., are explained in linguistic terms (Herskovits 1987) but not formally defined such that they could be implemented. Most often an infinitely dense collection of points, as in point set topology or in Euclidian geometry, is assumed. Goodchild (1990) proposes a ‘geographic reality’ based on points and values for properties of interest at these points. Implementations can only deal with explicit representations for a finite number of objects, thus a discretization is necessary to reach an implementable data model.

We therefore reach the following intermediate conclusion, namely to differentiate between three notions:

- Data structures (specifically geometric and spatial data structures):
  Detailed and low level descriptions of storage structures (traditional data structures) and the pertinent operations, with details of how the desired effects are achieved. They will not only provide a specific function (i.e. fulfill the conditions of an operation) but also are fixed in terms of performance, storage utilization etc. - they are a specific solution for a generic problem.

- Data models (specifically geometric data models):
  A comprehensive set of conceptual tools to be used to structure data. They are defined formally and are constructed such that they can be implemented.

- Concepts (specifically spatial concepts and geometry):
  Ideas, notions and relations between them which are used by humans to organize and structure their perception of reality. They differ depending on the task at hand, the circumstances and the experience of the persons. They are either informal, i.e. not formally defined or not (currently) definable, or formally defined but not implementable, due to the fundamental restrictions of computer systems (e.g. the fact that they are finite machines).

EXAMPLES OF SPATIAL CONCEPTS

In the past (Frank 1987) we have attempted in theoretical studies to define the data that describe the non geometric properties of geographic information. It is sufficient to abstract all attribute data to a vector of values of unspecified type, and no further interactions between specific operations on this vector and the spatial data need to be considered (geometric and non-geometric data in Gueting, 1988). This provides a base level description of spatial data.

If we structure the data in entities, there may be some additional structure between the entities (e.g. sets of all parcels belonging to a person, an ordered list of all schools in a district according to their capacity etc.). These are non-spatial aspects and have to be dealt with with the regular tools of the (non-geometric) data model. We will see that it is sometimes useful to base a geometric data model on a generic one (e.g. efforts have been made to model a cell based geometric data model using the relational data model (Gueting 1988) and to map geometric operations to operations on the generic data model.

A tentative set of spatial concepts are discussed in the next subsections. This list is not yet complete, and it is not even clear if a complete list is possible. There are other important spatial concepts, which are not included for various reasons; chief among them is a lack of clear understanding. A traditional view is to differentiate between an entity based view - space is constructed from objects that fill space - and a space oriented view, where each point in space has some properties. This view is philosophically well established - it can be considered to go back to Kant on the one hand and to Descartes on the other. This is a very important, theoretical as well as practical differentiation, which leads to a number of different concepts and differences in the operations applicable.

Sets of points

Space is thought of as a collection of an infinite number of dimensionless points which form a continuum. Each point is identified by a coordinate value (mathematically this is equal to R x R for 2 dimensional space) and this model assumes that the space is continuous and that the distribution of points is dense overall. For each point - at least theoretically - there exists a vector of attribute values that describe its properties. This is essentially the spatial concept that Goodchild mentions as ‘geographic reality’ (Goodchild 1990).

Thematic layers, surfaces

An attribute associated with space can be thought of as a continuous surface (with a single value of the attribute per point in space). This concept is used primarily for the topographic surface of the world but can be applied to other data. We may or may not assume that the surface is smooth and continuously differentiable, or the values change at some boundaries abruptly.

Euclidian geometry

Euclidian geometry is an entity oriented spatial concept. The objects it deals with are points and infinite lines, and the operations on them are explained by a set of axioms. There exists a mapping to coordinate space, with algebraic expression substituting for the Euclidean constructions with ruler and compass. Each point is represented by a pair of real numbers and formulae that correspond to geometric operations are given. The basic foundation of this model is thus very similar to the point set model, but Euclidean geometry structures space into discrete entities.

Partitions

A division of space in areas, such that all the areas sum up to the whole and no two overlap (i.e. they are pairwise disjoint) is often used. Subdivision of land into ownership parcels is thought of in this way, but also soils classifications are constructed following this concept. Mathematically, such a construction is known as partition. Practically, we find partitions that are constructed based on attribute values, i.e. the (connected) set of all points with a given attribute value (or a value in an interval, or set of values) and this leads to disjoint areas. These partitions are called ‘categorical coverages’ (Beard 1988). On the other hand, one often uses choropleth maps, which are partitions which were previously constructed, e.g. following political boundaries, for reporting census and similar statistical values (Robinson et al. 1984).

Delimited spatial entities

In lieu of partitions, one may just define spatial units, each with its boundaries, without enforcing that they be disjoint (i.e. without ‘planar enforcement’, Goodchild 1990). This concept is more of importance for conceptual reasons than for actual data collection, where the demand for completeness of data collection (one of the attributes of data quality, Robinson and Frank 1984) forces automatically a partition concept.

Cell topology

Cell topology is another, mathematically based concept, related to the continuous space concept. In cell topology, we deal with cells, of dimension 0 (points, so called 0-cells), of dimension 1 (arcs, so called 1-cells), dimension 2 (areas, so called 2-cells) etc. We are primarily concerned with relations between these objects, the boundary and co-boundary relations: an arc bounds an area, an area is bounded by (co-bounds) an arc; the same for arcs, which are bounded by points. In pure topology, the exact spatial location of nodes and arcs is not important, solely the spatial neighborhood is relevant. Thus configurations may be changed, as long as no cutting, hole puncturing etc. occurs.

Graphs

Graphs are built from two sets of objects, nodes and arcs and the connections between them, called adjacency. Variants of graphs have ‘directed arcs’. Graphs need not be planar (i.e. arcs may cross without being connected). There is a substantial set of algorithms known to compute properties of graphs. Graphs seem to be a good approximation to the concepts used for navigation with cars (where we have to follow roads, which form a graph) and other transportation problems, where a network of possible connections is given. A variant of great practical importance is the network, where individual points on the arcs can be addressed (for example by distance from one of the nodes, Goodchild 1990).

Cognitive spaces

It is - so far - not clear what are the exact properties of the cognitive concepts people use to deal with space. Observing problems with extending concepts gathered from ‘small scale spaces’ to other situations, Zubin proposed tentatively a set of spaces

(Mark, Frank et al. 1989), which reach from a more Euclidian view to a more graph oriented one:

Omniperspective: The small space one can perceive, where the mind's eye sees the object (e.g. a cup on a table) from all sides, even if only one side is actually visible.

Monoperspective: The case where a view of a space is collected from various glances and the connected view of space is constructed in the mind (e.g. a room).

Scene: Single perspective, where one sees only one side of an object and cannot infer its other sides (e.g. the perception of a building from the street curb).

Territory: The navigational concept, where one forms a concept of space by combining various views and experiences from interaction with the space (e.g. a town).

Imaging schema

Johnson provides a clear statement of how an image-schemata-based model of cognition would operate:

> "...Much of the structure, value, and purposiveness we take for granted as built into our world consists chiefly of interwoven and superimposed schemata...My chief point has been to show that these image schemata are pervasive, well-defined, and full of sufficient internal structure to constrain our understanding and reasoning. To give some idea of the extent of the image-schematic structuring of our understanding (...), consider the following partial list of schemata, which includes those previously discussed:
>
> Container
> Restraint Removal
> Path
> Near-Far
> Full-empty
> Contact Process
> Balance Compulsion
> Enablement
> Link
> Scale
> Matching
> Surface Object
> Blockage
> Attraction
> Center-Periphery
> Part-whole
> Superimposition
> Collection
> Counterforce
> Mass-Count
> Cycle
> Merging Splitting
> Iteration
>
> This brief list is highly selective, but it includes what I take to be most of the important image-schemata. If one understands 'schema' more loosely than I do, it might be possible to extend this list at length." (Johnson 1987 p.126).

GEOMETRIC DATA MODELS

The spatial concepts are typically not directly implementable, because they are assuming an infinite set of points (or another form of the same continuum assumption) and must be discretized. Discretization as the major modelling step is commonplace in geography (Goodchild 1990), but it is often just thought of as sampling and averaging over regular raster cells. Another limitation of spatial concepts is that some of them are not formalized, but just loosely described in terms of cognitive processes and experiments.

A geometric data model must have a well defined set of objects and operations on these objects. This fulfills the 'formal definition' requirement. The set of object (instances) must be finite, in order for the model to be implementable on a finite computer system. The behavior of the model is stated in terms of the effects of the defined (change) operations, which are observable with the given (observed) operations (Guttag, Horowitz et al. 1978).

To illustrate, there follows a short list of geometric data models and their characteristics as found in a geographic information system.

Regular tessellations of space (raster)

We can model continuous space by a finite set of small, regular shaped areas that tessellate it. This is a simple and useful method to discretize space, either by regular sampling, which determines the value for a specific location, or by averaging over the area involved.

Point sets using interpolation

We can record the value of an attribute at specific points, either regularly spaced on a grid or irregularly distributed and then provide an interpolation method which determines value for all intermediate points. There are a large number of variations on this theme, depending on what arrangement of points is permitted and what interpolation methods are assumed.

Spaghetti

Spatial concepts may represented by simple lines - usually this model is connected to a cartographic modelization, which represents reality as a map and the data model is then used to represent the map (and thus indirectly reality). The lines themselves may be modeled by a sequence of points, thought of as connected by straight lines, or more sophisticated interpolation methods may be selected.

Graph

The graph concept can immediately be translated in a data model. In order to simplify implementation, restrictions are often imposed, which may include planarity of the graph.
The concept can also be extended:

- the edges are directed;
- locations on the edges are possible without introducing new nodes;
- connections on the nodes are not all equal (i.e. there is an internal graph in the nodes, which need not be planar - so called turn tables); The connections between the arcs can be thought of as straight or may have detailed and determined form (again more or less restricted, depending on the implementation).

Topological data model

This model includes the topological concepts as well the partition concepts, as it appears difficult to implement a partition structure without the use of topological relations. The model is often restricted with limitations on:

- the form of the edges (often just straight, or approximated by arc of circle or splines);
- the number of nodes per cell;
- permission to create islands in cells or not.

Restricting the model to form a cell complex eliminates isolated nodes and edges which do not separate areas. The definition of operations on cells becomes much simpler if we demand that the cells form triangles forming simplicial complexes (in lieu of the more general cell complex) (Frank and Kuhn 1986).

Future research

We see that the data models, even if they can be reduced to a small number of typical ones, differ between implementations, because details of the implementation are allowed to 'show' at the conceptual user interface. This is usually justified by better performance. However, these small differences are costly, as they hinder transfer of data between systems and generally communication between systems and their users. The proposed geographic data exchange standard (Moellering 1987) informally defines a number of concepts which can be used to form a geometric data model (i.e. terms like node, arc, polyline), but the exact meaning of these terms cannot be given without the framework of a formal, algebraic definition using operations.

It is an attractive and important research plan to define these geometric data models formally, i.e. as algebraic specifications (Goguen 1989). From such a definition it can be shown how the concepts relate to each other. It is essential that we define mappings between these algebras, i.e. morphisms with map objects and operations (see Herring 1990; Mark, Frank et al. 1989). To a certain extent, Goodchild (1990) attempts to show how objects of one model can be deduced from another one - implicitly proposing the point set concept as general.

GEOMETRIC DATA STRUCTURES

There exists a large number of data structures, defined in more or less detail, to implement the geometric data model. Indeed, in the past it often seemed that one found first a geometric data structure which then implicitly defined a geometric data model. The geometric data model however should be the abstract view of the geometric data structure, not the other way round.

In order to see the difference between data model and data structure, one can simply observe that:

- data structure is concerned with performance, storage utilization and other implementation details;
- the data model is concerned with function.

If we formally describe the geometric data model, it is possible to test to what degree a data structure implements a model. In principle, one should not need to know the implementation details, and one implementation should be exchangeable for another one. The data structure should export exactly the operations defined in the model.

In the following we will only cite a few of the major data structures, without details as there exist an enormous number of variants for each of them.

Raster data structures

These implement the regular tessellation models. Implementation can be as straight array data structure; methods applicable to sparse arrays may work but best results are generally attained with methods to exploit spatial autocorrelation. The best known methods are run length encoding and hierarchical storage schemes, known as quadtrees (Samet 1989b). Holroyd (1990) discusses the problems of compression methods extensively.

Point sets

Data structures to store individual points can use either a tabular structure (and possibly some indexing methods for access) or exploit regularity in the distribution of the points in space, such that the location of a point can be inferred from the identifier (which often is directly mapped to a storage location and only implicitly represented). Implementation of interpolation methods differs widely and there is extensive literature on different interpolation methods and how they are best carried out. The choice evidently depends also on the field of application, as some methods are better able to deal with certain special situations.

Topological data structures

The basics of implementing a topological data structure are well understood, but there are considerable differences between them. They differ in the exact data model used and in the details of the implementation. There is considerable literature on the subject in CAD/CAM, but a definite text is lacking.

GEOMETRIC DATA STRUCTURES USED FOR INDEXING

In most applications that store spatial data, access to the data is not only based on identifiers (e.g. parcel numbers, names of towns etc.) but also on spatial location. One needs to answer questions such as ‘what is at location x,y?’ or ‘find all objects inside a window’. The data model for this problem consists of data objects for which a spatial location and extent is defined in a coordinate system, and access operations retrieve all objects within a window; or finds the closest neighbor object to a given object (Frank 1981; Frank 1983b; Frank and Barrera 1989).

A large number of geometric data structures were developed specifically for this indexing purpose and a number of the data structures included above can be used as well. Buchmann, Guenther et al. (1990) give an updated overview of this interesting field; it is not the primary concern of this article, because the indexing structure as such does not participate in the modeling of reality. It contributes a performance gain over an operation which could be, in principle at least, executed without use of the indexing structure. It is possible to find all objects within a window just by sequential inspection of all stored geometric data objects. This is clearly impractical for most larger data collections, but this is only a performance issue, not a modeling one. From a practical point of view, it was found that a geographic information system should use a spatial indexing structure, but results from comparison of different data structures are difficult to generalize.

CONCLUSIONS

This paper began with an examination of the use of the terms ‘data model’ and ‘data structure’ in the computer science oriented database literature. ‘Data model’ means a set of conceptual tools to describe the logical or conceptual structure of the data, whereas ‘data structure’ is used to describe a specific implementation of a data model. A data model describes on the abstract level objects and their behavior, but only the data structures fix performance aspects like storage utilization and response time. It was found that similar concepts apply to modeling the geometric aspects in a geographic information system and it was proposed that the term ‘geometric data model’ should be used to describe an abstract view of geometry and geometric properties of objects. It is recommended that this be formalized using an object-oriented viewpoint as an algebraic structure with a set of objects, operations to construct, change and observe these objects and axioms (rules) which explain the result of the operations in terms of other operations. Geometric data models must be formally defined and it must be possible to implement them on a current computer system. Geometric data structures are then specific implementations which provide the operation demanded in a geometric data model, using specific storage structures and algorithms. Data structures exhibit specific performance properties, storage utilization and speed of operations being the most important ones. They are optimized for certain cases and yet may not be suitable for other applications.

Unlike administrative applications of databases, geographic information systems model reality, or the elements of reality humans perceive. In order to understand and structure their spatial perceptions, humans seem to use more than a single concept of space, and these concepts often are either not formally defined or not able to be implemented. The term 'spatial concept' is used to describe these notions, which are then formalized and often discretized to form a geometric data model.

A comprehensive description and comparison of geometric data structures is the next major goal. It is hoped that the large number of data models which are heavily driven by implementation, can be reduced to a smaller number of fundamentally necessary traits, for which implementations can be found. This would make comparison of actual systems, communication between geographic information system users and transfer of data between systems much easier as one can then use the reference data model and not be concerned with the conceptually irrelevant differences in the implementations.

REFERENCES

Alexandroff, P., 1961. *Elementary Concepts of Topology*. Dover, New York.

Beard, K., 1988. Multiple representations from a detailed database: a scheme for automated generalization. Unpublished PhD Thesis, University of Wisconsin - Madison.

Birkhoff, G. and J.D. Lipson, 1970. Heterogeneous algebras. *Journal of Combinatorial Theory* 8:115-33.

Buchmann, A., O. Guenther, T.R. Smith and Y.-F. Wang, eds., 1990. *Design and Implementation of Large Spatial Databases*. Lecture Notes in Computer Science No. 409. Springer Verlag, New York.

Chan, K.K.L. and D. White, 1987. Map algebra: an object-oriented implementation. *Proceedings, IGIS: The Research Agenda*. NASA, Washington DC.

CODASYL, 1971. *CODASYL Data Base Task Group Report*. Conference on Data System Languages,ACM, New York.

Codd, E.F., 1970. A relational model for large shared data banks. *Communications of the ACM* 13:377-87.

Codd, E.F., 1982. Relational data base: a practical foundation for productivity. *Communications of the ACM* 25(2):109-17.

Corbett, J.P., 1979. *Topological Principles of Cartography*. Report No. 49. Bureau of the Census, Department of Commerce, Washington DC.

Date, C.J., 1986. *An Introduction to Database Systems*. Addison-Wesley, Reading, MA.

Diaz, B.M. and S.B.M. Bell, 1986. *Spatial Data Processing using Tesseral Methods* (collected papers from Tesseral Workshops 1 and 2). Natural Environment Research Council, Unit for Thematic Information Systems, Reading.

Dittrich, K.R., 1988. Advances in object-oriented database systems. *Proceedings, Second International Workshop on Object-Oriented Database Systems, September 27-30, Bad Munster am Stein-Ebernburg, FRG*. Springer Verlag, Berlin and New York.

Dutton, G., 1979. *First International Study Symposium on Topological Data Structures for Geographic Information Systems*. Harvard Papers on Geographic Information Systems. Harvard University, Cambridge, MA.

Frank, A.U., 1981. Applications of DBMS to land information systems. *Proceedings, Seventh International Conference on Very Large Data Bases (VLDB), Cannes, France*

Frank, A.U., 1983a. Datenstrukturen fur Landinformationssysteme - semantische, topologische und raeumliche Beziehungen in Daten der Geo-Wissenschaften (Data structures for land information systems -semantical, topological and spatial relations in data of the geosciences). Unpublished PhD thesis, Swiss Federal Institute of Technology, Zurich.

Frank, A.U., 1983b. Storage methods for space related data: the field tree. *Proceedings of Spatial Algorithms for Processing Land Data Symposium*. Lincoln Institute for Land Policy, Cambridge, MA.

Frank, A.U., 1984. Computer assisted cartography: graphics or geometry? *ASCE Journal of Surveying Engineering* 110(2): 159.

Frank, A.U., 1987. Overlay processing in spatial information systems. *Proceedings, AutoCarto 8, Baltimore*. ASPRS/ACSM, Falls Church, VA.

Frank, A.U. and R. Barrera, 1990. The fieldtree: a data structure for geographic information systems. In A. Buchmann, O. Guenther, T.R. Smith and Y.-F. Wang, eds., *Design and Implementation of Large Spatial Databases*. Lecture Notes in Computer Science No. 409. Springer Verlag, New York.

Frank, A.U. and W. Kuhn, 1986. Cell graph: a provable correct method for the storage of geometry. *Proceedings, Second International Symposium on Spatial Data Handling, Seattle*.

Frank, A.U. and D.M. Mark, 1990. Language issues for geographical information systems. In D.J. Maguire, M.F. Goodchild and D.W. Rhind, eds., *Geographical Information Systems*. Longman, London.

Goguen, J., 1989. Modular algebraic specification of some basic geometrical constructions. In D. Kapur and J.L. Mundy, eds., *Geometric Reasoning*. MIT Press, Cambridge, MA.

Goodchild, M.F., 1990. Geographical data modeling. Technical Report 90-11. National Center for Geographic Information and Analysis, Santa Barbara, CA.

Gueting, R., 1988. Geo-relational algebra: a model and query language for geometric database systems. *Proceedings, Advances in Database Technology---EDBT '88, International Conference on Extending Database Technology, Venice, Italy*. Springer Verlag, New York.

Guttag, J.V., E. Horowitz and D.R. Musser, 1978. Abstract data types and software validation. *Communications of the ACM* 21(12):1048-64.

Herring, J.R., 1990. TIGRIS: a data model for an object oriented geographic information system. Presented at GIS Design Models and Functionality, Leicester, March.

Herskovits, A., 1987. *Spatial Prepositions in English*. Cambridge University Press, Cambridge.

Holroyd, F., 1990. Raster GIS: models of raster encoding. Presented at GIS Design Models and Functionality, Leicester, March.

Johnson, M., 1987. *The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason*. University of Chicago Press, Chicago.

Knuth, D.E., 1973. *The Art of Computer Programming*. Addison-Wesley, Reading, MA.

Lakoff, G., 1987. *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*. University of Chicago Press, Chicago.

Lakoff, G. and M. Johnson, 1980. *Metaphors We Live By*. University of Chicago Press, Chicago.

Mark, D.M., 1989. Cognitive image-schemata for geographic information: relations to user views and GIS interfaces. *Proceedings, GIS/LIS '89, Orlando*. ASPRS/ACSM, Bethesda, MD.

Mark, D.M., A.U. Frank, M.J. Egenhofer, S.M. Freundschuh, M. McGranaghan and R.M. White, 1989. *Languages of Spatial Relations: Initiative Two Specialist Meeting Report*. Technical Report 89-2. National Center for Geographic Information and Analysis, Santa Barbara, CA.

Moellering, H., 1987. *A Draft Proposed Standard for Digital Cartographic Data*. National Committee for Digital Cartographic Standards, Washington DC.

Neisser, U., 1976. *Cognition and Reality*. Freeman, San Francisco.

Peuquet, D.J., 1983. A hybrid structure for the storage and manipulation of very large spatial data sets. *Computer Vision, Graphics, and Image Processing* 24(1):14-27.

Peuquet, D.J., 1988. Representations of geographic space: toward a conceptual synthesis. *Annals of the Association of American Geographers* 78(3):375-94.

Robinson, A.H., R. Sale, J.L. Morrison and P. Muercke, 1984. *Elements of cartography*. Wiley, New York.

Robinson, V. and A.U. Frank, 1984. About different kinds of uncertainty in collections of spatial data. *Proceedings, Autocarto 7*. ASPRS/ACSM, Falls Church, VA.

Samet, H., 1988. Hierarchical representations of collections of small rectangles. *ACM Computing Surveys* 20.

Samet, H., 1989a. *Applications of Spatial Data Structures*. Addison-Wesley, Reading, MA.

Samet, H., 1989b. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA.

Tomlin, C.D., 1983. *Digital Cartographic Modeling Techniques in Environmental Planning*. Unpublished PhD Thesis, Yale University.

Tomlin, C.D., 1989. *Geographic Information Systems and Cartographic Modeling*. Prentice-Hall, Englewood Cliffs, NJ.

Tsichritzis, T.C. and F.H. Lochovsky, 1977. *Data Base Management Systems*. Academic Press, New York.

Ullman, J.D., 1988. *Principles of Database and Knowledgebase Systems Vol 1*. Principles of Computer Science Series. Computer Science Press, Rockville, MD.

Zehnder, C.A., 1981. *Informationssysteme und Datenbanken*. Verlag der Fachvereine, Zurich.