# Improving Spatial Analysis in GIS Environments

Geoffrey Dutton
*Spatial Effects*
150 Irving Street
Watertown MA 02172 USA
qtm@cup.Portal.Com

## Abstract

Current GIS technology tends to impede problem-solving for many of its users and is difficult for vendors to develop and support. Why this may be the case and what might be done about it is explored in this paper. Three problems shared by most commercial GIS' are identified and examined: inadequate data models; inferior application development tools; insufficient on-line expertise. It is argued that each factor inhibits robust spatial analysis, and limits the usability of analytic and cartographic GIS outputs. Suggestions based on recent research directions and emerging software engineering practices are given for addressing deficiencies in these three realms. Certain properties of and synergies among data models, application toolkits and expert systems are explored as keys to improving data and knowledge management, user interaction and spatial analysis in geographic information systems.

## Introduction

Geographic Information Systems (GIS) are a protean emerging technology involving many primary data sources (spatially sampled measurements of the natural and human environment, surveying and photogrammetric data, digitized maps and remotely-sensed images), diverse data structures (points, polygons, networks, rasters, quad - or whatever - trees), complex databases (geometric, topological, attribute and metadata, relational, hierarchical, distributed and hypermedia), evolving analytic methods (network and surface synthesis and analysis, feature extraction, spatial overlay, temporal change and other attribute analysis), high-quality cartography (2+D rendering, what-if graphics, engineering plans, thematic maps) and high sensitivity to data quality (positional accuracy, feature coding, resolution and scale effects, and spatial/temporal aliasing). GIS emerged from laboratory gestation in the mid-1980's to confront an explosion of environmental challenges and applications. But while computing hardware capable of manipulating complex spatial data is increasingly within the reach of users, GIS developers do not completely agree about how spatial software should best be built, spatial data structured, applications wrought and spatial analysis conducted. This is in part due to a lack of high-level tools, but also is a consequence of relying upon low-level constructs that are proving increasingly inadequate. Twenty-five years of GIS progress should not prevent us from re-evaluating our basic assumptions and prevailing models. Doing this might broaden our perspectives and invigorate our technology.

## Solving spatial problems

GIS evolved from early attempts to conduct spatial analysis using digital cartographic data in vector and raster form. Success has always been limited by the amount of information encoded into cartographic databases. Sets of points, lines and polygons, while fully defining entities in a "geographic matrix" (Berry, 1964), fail to model spatial relationships among them. To such descriptions topology has been added (Corbett, 1977), defining how cartographic entities connect to one another. Other, potentially valuable capabilities not now in use have been proposed: data quality documentation (Chrisman, 1984); global hierarchical spatial indexing (Dutton, 1989; Goodchild and Yang, 1989; Fekete, 1990); temporal data management (Langran, 1989). While these approaches could be incorporated into existing systems, the effort and cost required would be formidable; a new generation of software may be needed instead.

In any case, alternatives to current spatial data models are already needed. In the author's view, if academia and industry are to meet the challenge of supporting global environmental science, developers will have to retool GIS databases at a rather basic level. This is because so many of the "coverages", "partitions" and "projects" by which GIS's administrate databases are modeled as planar, cartesian chunks of the world, represented as maps. Although many systems can perform transformations between projections and into latitude and longitude, this is usually only done to "register" coverages by mapping coordinates into a preferred planar projection. Certain special-purpose and in-house GIS's store spherical coordinates in their databases, and thus are in principle capable of working on a global scale. But the mathematics involved in manipulating such data can be costly, and the ambiguities inherent in attempting to positively identify points and their loci will continue to confound applications, especially when data quality information is lacking or goes unused.

Despite the fact that most vendors heavily promote "solutions" (sets of niche applications) GIS isn't really the sum of vertical markets; it's a technical infrastructure (like DBMS) upon which applications may be erected. While different applications of spatial data have unique if not conflicting analytical requirements (what methods do crop assessment and network analysis have in common, for example), their implementation usually insures that their data will remain incompatible. It may be that geographic data deserves support at the system level, commensurate to the facilities hardware vendors now provide for manipulating text, numbers, tables, images, abstract datatypes and user interfaces. Given sufficiently capable data structures, methodologies, and advice packaged in forms accessible to end-users, more robust and specialized GIS applications could be generated more easily; these would effectively combine known spatial analytic and cartographic methods with canonical ways of storing, retrieving and manipulating spatial data, guided by facts about data domains and rules that apply to them.

*Solutions beget problems*

In a recent paper describing a systems-level architecture for supporting use of hypertext within and across diverse applications, Kacmar (1989) identifies several problems that existing hypertext implementations exacerbate:

> Current hypertext systems have attempted to a provide an all-inclusive work environment for the user. However, few systems have been able to realize this goal. Thus, users are required to utilize several applications for their activities and must enter and exit applications in order to accomplish specific tasks. The hypertext system becomes yet another application and other user interface mechanism which must be learned and used. (Kacmar, 1989, p. 98)

One can substitute "GIS" for "hypertext system" in this text without changing its sense, just its context. In contrast to hypertext authors, however, GIS users must be more than casually aware of the nature of the database and data structures their systems manipulate, due to the various special properties of spatial entities that they model (hypertext data mainly consists of text fields linked as a semantic net having abstract, user-imposed and self-specified "spatial" relationships). That is, while all the semantics of a hypertext database are user-specified, much of the semantics of spatial information is given or constrained by physical laws, common law, administrative regulations, data structures and measurement theory. When a GIS is used to model spatial semantics, the ways in which rules are applied and information is communicated to users tend to vary greatly in completeness, consistency and complexity.

When enhancing their systems, GIS vendors tend to maintain compatibility with earlier versions (to safeguard users' investments), even though it might be technically advisable to radically redesign applications. One result is that new commands or modules tend to be added on top of or alongside of existing ones; this can steepen the learning curve for affected applications, and still not assure that the tools provided will serve users' purposes. Only highly-motivated users may exercise the more complex applications and options, often to discover that useful features or parameters in enhanced applications aren't available in other, related contexts. As GIS data grows more complete and complex (that is, as systems incorporate more information about spatial semantics), the "span of control" confronting users will also increase, and vendors will have to work hard to make systems uniformly and consistently usable. This will be necessary regardless of what type of user interface is involved (command lines, menus, direct manipulation, hypermedia).

*Problems (sort of) fade away*

Like hypertext, GIS is not yet a mature technology. This should not, however, be used as an excuse for perpetuating difficulties involved in learning and applying GIS. Cooke (1989) argues that we are about to enter the "post-GIS era", in which the bulk of data capture activities will have already been accomplished or become relatively automated. In such a milieu, our attention will naturally turn toward modeling and analyzing spatial phenomena; much of the output from GIS will be non-graphic (such as inventories of property, estimations of resource acquisition and operating costs, environmental status of specified areas, or the address of the nearest elementary school), and users will demand error reports, confidence limits and sensitivity analyses for data they analyze. They will also need assistance in browsing through ever-larger spatial inventories, in formulating queries to extract data relevant to their purposes and in specifying the steps necessary to perform particular analyses. Still, progress will inevitably occur, in Cooke's view, leading to a flourishing of environmental applications:

> Technical issues of digitizing, coordinate conversion, map-edge matching and topological editing will fade into history. We finally will be able to turn our creative energies to solving real problems, of which we have plenty. Pollution, ozone depletion, the greenhouse effect, all are exacerbated by the inefficient logistical operations resulting in unnecessarily burning fossil fuel. (Cooke, 1989, p. 55)

But will data maintenance tasks and data quality problems ever fade away, and if so, what will cause this to happen? Major users of GIS and CADD, such as local governments, seem unable to keep up with the pace of change in their jurisdictions, and it is hard to imagine that their digitizing activity will ever cease. New versions of TIGER and other base files will be issued, but it may never become trivial to integrate them with an organization's database. It seems apparent that spatial data is not going to get easier to handle just because it is growing more complete and accurate, and can be obtained in digital form, even though much drudgery may be eliminated for users. Integrating spatial data isn't inherently difficult; it has been made difficult by a plethora of local coordinate systems, differing (and inadequate) data models and data structures, primitive data interchange standards, insufficient data quality information and a disinclination to use it. Metaphorically speaking, we have built a maze of datatypes, tools, techniques and topology, and are getting frustrated because we can't find our way out. Perhaps a good sales slogan for our industry would be "Lose yourself in GIS (it's easy)!"

## Retooling GIS

To save GIS from crippling itself and to help users meet the challenges that their work presents, GIS researchers and developers need to take a critical look at the factors that limit the effectiveness, reliability and usability of current technology. In the author's view, such reexaminations should focus on three problem areas, which seem to map to three scales of software engineering and involve three groups of actors. These key problem areas, or realms, are:

| | |
|---|---|
| 1. Data models; | How spatial data is represented for computation |
| 2. Application toolkits; | Better ways of constructing custom software |
| 3. On-line expertise. | Access to knowledge about data and methodology |

Table 1 describes an operational context for these realms.

*Table 1*

**Three Critical Realms of GIS Technology and Application**

| REALM | SCALE | ACTORS | CURRENT PROBLEMS |
|---|---|---|---|
| Data models | Micro | Researchers | Need to improve, standardize |
| Application toolkits | Meso | Developers | More flexible, customizable |
| On-line expertise | Macro | Users et al | Better help, reports, advice |

"Scale" refers to the extent of software modules that reify the realms (e.g., functions, libraries, subsystems). "Actors" are those professionals who are most central to implementing the realms. As many GIS professionals have played all the above roles at one time or another, they should be aware of the impact of each of the realms on actors' activities. This is not to exclude the effects of related topics (such as object-oriented software, numerical algorithms, hypermedia and user interface design) on the state of the art. But these and other realms are receiving attention not only in the GIS community but in software engineering in general. The three realms listed in Table 1 pose direct challenges to GIS, and will have to be addressed by both our industrial and research enterprises.

*Data Models*

Most GIS employ data models (the conceptual organization used for spatial and aspatial data) inherited from computer cartography, CADD and civil engineering, image processing and management information systems. They tend to be, as Cooke (1989) points out, based on the "map-as-graphic" paradigm rather than "map-as-database". The variety of data models (e.g., image, object, network, layer) and many variations in the data structures used to implement them (arc-node, grids, quadtrees, TINS) has led to difficulties in comparing and exchanging datasets, even when "standard" interchange formats are used; some relationships may not be encodable, hence are lost, unless they are later reconstructed — often at great expense. It has also bred a somewhat cavalier attitude that encourages incompatible data structures to coexist, even within the same database. Thus, the decades-old "vector v. raster" debate, having never been resolved, has been made moot by concessions that each structure has unique strengths; as one may always be converted to the other, there is no need to make a choice other than for tactical expedience.

While vector- and raster-encoded spatial data may in some sense be equally good (both are certainly useful), they can also be considered to be equally bad. Raster files tend to be bulky, unstructured and insensitive to variations in data density. Vector data structures can be cryptic and complex to manipulate, and often fail to express variations in data quality. Both models tend to ignore or filter out important aspects of spatial structure in abstracting geographic data. To compensate for these losses, ingenious and sophisticated methods have been built into GIS's: "fuzzy overlay", "rubber-sheeting", and resampling and filtering in both the spatial and spectral domains. But rarely are the tribulations that these methods are designed to overcome traced back to their source: loss of information in exchanging, digitizing and scanning maps and images due to impoverished data models.

Remote sensing technology has succeeded in recovering remarkably useful data from rather imperfect platforms, and has developed a canon of tools and techniques for correcting geometric and radiometric errors and finding structure in image data, often implemented as black-box functions which are tricky to integrate and easy to misuse.[1] Cartographic digitizing methods have also improved, especially in terms of avoiding, identifying or compensating for operator blunders and errors (White and Corson-Rikert, 1987). Most GIS's still express the quality of digitized data rather simplistically, relying on a few global parameters (such as U.S. map accuracy standards express), which fail to express local variations in spatial uncertainty inherent to the phenomena being captured. Even if this information were to be provided, prevailing GIS data models tend to have no place to put it, and their analytic procedures generally make little use of data quality information in their deliberations.

This state of affairs represents an ironic twist of autocarto evangelism[2]: After SYMAP and other software enabled digital thematic mapping, a lot of effort went into explaining to cartographers what polygons were, and this eventually instilled in them an abiding attachment to coordinates, which for awhile they resisted, then embraced just around the time they were told that polygons weren't enough, and they needed to learn about topology. After DIME (A.D.) embedded map networks in a rigorous mathematical framework (Cooke and Maxfield, 1967), it seemed for awhile that topology would solve most spatial data-handling problems, because coverages could now be verified to be complete and correct. The naîvete of this presumption was made evident by the advent of GIS; as soon as analysts began to merge and overlay

[1] Beard (1989) identifies *use error* as the "neglected error component" in GIS applications. While human error is difficult to quantify, it seems apparent that the more commands, options and parameters that a user confronts at a given moment, the greater the likelihood that (s)he will make a mistake.

[2] Many highlights and sidelights of the development of computer cartography and GIS are related in a recent issue of *The American Cartographer* (vol. 15, no. 3, July 1988), subtitled "Reflections on the revolution: the transition from analogue to digital representations of space, 1958-1988".

map data derived from different sources they found that while their computer could connect complex mazes of dots and lines into a single network and name all the objects therein, many if not most of these often turn out to be artifacts that have no basis in reality.

Today, c. 24 A.D., the source of such hassles is widely acknowledged to stem from failures to maintain data quality information within spatial databases. What is *not* as widely appreciated is that this may directly issue from twenty-five years of representing spatial locations as two- and three-dimensional coordinate tuples that have no inherent scale, only precision. This peculiar myopia has been termed *the fallacy of coordinates* (Dutton, 1989b), the (often unconscious) leap of faith that coordinates actually exist. It describes, but fails to explain, how entire professions and much software have come to accept point coordinates as if they were natural phenomena, like pebbles or protons. Still, it is not surprising that coordinates are reified in a culture which regards land as real estate, in which inches of frontage can cost dearly and where boundaries need not hew to visible landmarks. It is odd and rather distressing to have to preach the heresy that coordinates are the antichrist of spatial data handling, given that computer cartography and GIS have been around for a quarter of a century. But there are times in any walk of life when conventional wisdom bears reexamination.

Part of the reason why coordinates prevail is due to the view that digital spatial data represents maps, which in turn represent the world. It seldom seems to occur to GIS developers that they might better serve users by regarding maps as products of, rather than as the basis for their systems. As a result, most GIS's use "cartographic data structures" (Peucker and Chrisman, 1975), which are good at encoding features on maps, but which eventually fail to represent much of the evidence available about distributions of things and events on our planet. Failure to handle temporality is one resultant problem (Langran, 1989), but there are others. Goodchild (1988) offers an example of how technical factors have shaped and constrained the development of computer cartography and GIS:

> In the case of forest inventory maps, the need for accurate inventory clearly overrides any question of cartographic clarity and ease of perception. However forest inventories continue to be mapped using bounded areas to portray homogeneous forest stands, suggesting in this case that technological constraints, specifically the inability to show transition or heterogeneity, have outweighed any more abstract cartographic principles. ... The consequences of those constraints can be rationalized as intelligent choices, and are so fundamental that it is difficult to consider alternatives, but they are in actuality severely restricting, and influence not only the way we portray the world but also the way we observe it. (Goodchild, 1988, ps. 312 & 317)

While acknowledging that any data model has limitations, Goodchild points to emerging alternative spatial data structures which might overcome problems inherent in electronic emulations of pen-and-paper technology:

> In one sense, [quadtrees] represent a departure from fixed scale in the form of fixed pixel size in rasters or fixed levels of spatial generalization in vectors. They are non-intuitive in that they correspond to no conventional pictorial view, but have meaning only as digital representations. In quadtree data structures we are beginning to see the emergence of a genuinely new technology in which methods have no obvious conventional analogues. At the same time the constraints imposed by the technology are radically different. (Goodchild, 1988, p. 316)

It is worth stressing again that limitations of map-as-graphic and point-line-area paradigms cannot be overcome without purging ourselves of the notion that locations in the real world are dimensionless points; neither will we make real progress by continuing to pretend (at least in our databases) that the Earth is flat and that we occupy its lower left-hand corner. The former prejudice prevents us from modeling spatial distributions in ways that capture their indeterminate and scale-dependent qualities. The latter assumption inhibits development of GIS databases and techniques that can deal with information from diverse sources and operate at continental or global scales. Such databases are being built (often haphazardly) at accelerating rates, and global GIS issues can no longer be swept under cartesian rugs. According to Tomlinson:

> The ability to integrate data with a variety of formats (raster, vector, street address and tabular) from different sources, at different levels of reliability, at different scales, by people with different skills, using different computers, in different countries, connected by communication networks, is a very real requirement in the foreseeable future. (Tomlinson, 1988, p. 259).

One recent approach to meeting these challenges is to design spatial databases that represent locations hierarchically, by indexing to positions occupied by vertices and faces of nested polyhedra, successively approximating the surface of a planet. While polyhedral map projections are not new (going back at least to the time of the artist Albrecht Dürer), the idea of tessellated, polyhedral data storage hierarchies is probably less than a decade old and still relatively unexplored (Peuquet, 1988); few schemes specific to GIS have been proposed (Dutton, 1984[3], 1989a[4]; Goodchild and Yang, 1989[5]; Fekete, 1990), and none of these have been demonstrated in fully operational contexts.

While not nearly enough research as been undertaken to verify expectations, one can anticipate a number of benefits that might flow from implementing such data models. Casting coordinates into hierarchical planetary tessellations might mitigate many of the data-handling problems that plague current GIS technology and short-circuit spatial analysis: using such methods, any planetary location can be canonically encoded, regardless of where it is or the resolution at which it is identified; features could be modeled more readily, matched and integrated more easily, with certifiable accuracy; different datasets encoding diverse locations could be merged with greater confidence and ease. Multi-resolution storage lets primitive data elements specify their inherent accuracy; collections of such elements can model complex objects, which can be retrieved at a variety of appropriate scales. Such collections may be cast into both raster and vector formats, but will have other

---

[3] The *geodesic elevation model (GEM)* is a dual (cube-octahedron) polyhedral tessellation designed to encode terrain relief of planets using two alternating ternary hierarchies. This horizontal organization was coupled with difference-enoding of elevations to provide a compact, self-calibrating and scale-sensitive representation of topographic relief. *GEM* was simulated, but never implemented.

[4] The *Quaternary Triangular Mesh (QTM )* scheme derives from *GEM*. *QTM* is a region quadtree composed of triangles. It represents a planet as an octahedron comprised of 8 quaternary triangular grids, and can encode locational data both as hierarchies and sequences. Collections of such codes can be structured to represent geographic objects at specific scales. *QTM* has fractal properties, which may be exploited by modeling locations as basins of attraction (*attractors*), hexagonal regions centered on *QTM* grid nodes and composed of six adjacent triangular tiles, to which all locations in their domain alias at some level of detail. Attractors knit together adjacent *QTM* domains, and may aid in identifying and preventing "slivers" when overlaying vector-encoded map features.

[5] Goodchild and Yang's *Triangular Hierarchical Data Structure (THDS)* is a variation of *QTM*. What distinguishes *THDS* from *QTM* is the simplicity of (a) its facet numbering scheme (which ignores attractors), (b) its geodesic computations (all subdivision occurs on planar octahedron facets), and (c) its indexing algorithms (although transformations between the two orderings have been developed).

qualities (deriving from the properties of recursive geodesic tessellations) that may be exploited by new species of data structures and algorithms.

*Application Toolkits*

Spatial analysis tasks range from primitive computations such as nearest-neighbor identification and interpolation of point sets to multi-stage simulations of urban growth and multi-layer land suitability studies. Analysts often engage in *ad hoc* explorations of data before (or in lieu of) settling down to a standard methodology. Many vendors hasten to fulfill requests from users for new analytic capabilities, resulting in ever-greater arrays of GIS applications, commands and options. Some systems let end users cobble together the functionality they need as procedures coded in a macro language provided by the vendor, similar to the way most spreadsheet applications are built. Larger vendors have application consultants who extend existing applications by writing source code modules and linking them in as new commands. Such customization activity is commonplace for any number of reasons, including:

- GIS, like CADD systems, must operate in many diverse environments;
- Many spatial analysis procedures are sensitive to details of data models;
- Data items (e.g., "attributes") may reside in various foreign databases;
- GIS applications and analytic methodologies are still evolving.

As GIS's and their applications proliferate, the pace and economics of software development make it difficult for vendors to keep up with analysts' demands for new functionality, and new users face steeper learning curves, requiring increasing amounts of documentation, training and time in order to become productive. In such a milieu, costs of developing and enhancing applications tend to increase, regardless of who performs the work. These seem to be problems generic to all applications software systems, and are by no means unique to GIS. They do, nevertheless, represent real impediments to improving GIS analytic capabilities.

Segments of the software industry have begun to adopt a variety of computer aided software engineering (CASE) tools to manage various stages of the software lifecycle, from requirements analysis and functional specification to user interface construction and code generation. In addition, end-user environments for building microcomputer applications are gaining in popularity, particularly those that provide direct-manipulation visual interactive programming (VIP) capabilities (Sabella and Carlbom, 1989). Many such tools are based on object-oriented (OOP), rather than procedural programming styles (Cox, 1987). Proponents of this approach maintain that it reduces the need for debugging, clarifies program structure, speeds development and increases software reusability. Even if these claims are true, OOP methodology may not help developers create better algorithms, not be as computationally efficient as traditional procedural code, and may prove hard to interface with the miscellany of data structures typically found in GIS environments.

The jury is still out concerning the utility of CASE, VIP and OOP to software developers and end-users. But it is apparent that many of the tasks traditionally assumed by developers and vendors are, by economic necessity and by popular demand, shifting to user domains. It also seems apparent that spatial analysis in a GIS environment is most effectively conducted when people most familiar with the problem domain actively participate in software development. This points to the provisional conclusion that GIS vendors ought to be encouraged to provide visual programming environments in which users can craft analytic software and other finicky functions, such as database interfaces, data filters and thematic maps. Such modules should be made as easy as possible to create, and this implies that objects in GIS databases need to represent themselves with appropriate data structures and manipulate themselves with appropriate methods, in transparent ways.

An informative account of modeling and visualizing geological reservoir data (Sabella and Carlbom, 1989) describes a set of tools called *Gresmod,* a laboratory testbed for rendering heterogeneous collections of geometric objects. In this prototype, users at a Xerox workstation worked with *Gresmod*'s OOP modeling environment and diagrammatic interface to specify a data base of geometric structures (curves, planes, and solids encoded as octrees, representing oil reservoirs), perform limited analysis, and render it in 3D on a high-performance graphic display. *Gresmod* was written in an OOP language called *Strobe,* which builds a set of knowledge bases for class hierarchies, attributes and methods. *Gresmod*'s direct-manipulation user interface (also knowledge-based), was built with the *Grow* toolkit for customizing the *Impulse-88* UIMS used by Sabella and Carlbom. *Strobe,* in turn, is implemented in Interlisp-D; it includes a graphic editor for objects, a run-time environment, and manages knowledge bases. The study provides an "object lesson" in how diverse forms of spatial data can be manipulated in unified fashion in a visual interactive environment. It shows how spatial analysis can be enhanced when tools for data modeling, application building and knowledge management are provided.

Commercial GIS technology can't do this yet, but is headed in these directions. Providing such environments will propel fundamental conceptual and architectural changes in system design. It is both technically possible and commercially expedient to transfer responsibility for application development to user communities, rather than continuing to place this staggering burden on vendors. This means that vendors must move away from offering *solutions* and toward providing *toolkits* that enable users to solve their particular, idiosyncratic problems. This approach is being adopted by vendors of high-performance workstations for data visualization (Upson *et al*, 1989); users may create or modify visual models by editing on-screen dataflow diagrams and property sheets, without the need for much if any program coding.

*On-line Expertise*

Printed and on-line documentation for most applications software packages generally include tutorials on how to get started, detailed command descriptions and one or more examples of data processing using sample datasets. As a GIS may include dozens of functions (ranging from map digitizing and topological editing, through feature and attribute definition, analytic procedures and display formatting, not forgetting project management activities), each of which may have dozens of commands, and its printed documentation can easily fill a four-foot shelf. While the general workflow at most GIS installations has many common and predictable aspects (digitize, edit, structure, extract, merge, overlay, analyze, report, map), the details of system configurations and users' data and applications differ enough to limit the utility of vendors' printed documentation and on-line help.

While users can normally refer to manuals to find information that answers certain types of questions ("What does command X do?"; "What commands do *Y* to data of type *Z*?"), they often can't find answers to many other kinds of questions ("What do I do next?"; "How will executing command *X* affect my database?"; "How much error might attribute *I* of object *J* have?"). The number of paths users can follow through a complex application is essentially unlimited. Certain paths may yield equivalent results with some data, but produce quite different results with other data; such discrepancies may derive from the content and quality of data being processed, parameters and options selected by users, the order in which commands are invoked, software bugs or be totally inexplicable and unreplicatable (known to hackers as the POM — Phase of the Moon — effect). And while the underlying database machinery may be able to roll back transactions that go awry, the amount of processing and operator time wasted in doing this can be a great drain on users (GIS transactions are typically quite lengthy, and all work performed between a check-out and a roll-back may have to be sacrificed, even if most of it produced correct results). While such unpleasant surprises may be the results of encountering software bugs or pathological data, most stem from the impossibility of grasping all the *normal* consequences of executing particular commands in particular sequences on particular sets of data.

The sheer complexity of working in a GIS environment is likely to increase as applications become more analysis-intensive. Printed and on-line documentation cannot even in principle address this problem, because most of the complexity is combinatorial, most of the questions context-sensitive. And while there are people who are skilled at dealing with such matters, such knowledge takes a long time to acquire and tends to be idiosyncratic and difficult to transfer to others. Even though the population of GIS gurus is growing, expertise of this type will remain a scarce resource for both vendors and users for the foreseeable future.

Many data-handling decisions are made automatically by software, based on options and parameters input by users (or their default values) and variables and constants stored in datasets (or their estimates). These are applied either on a case-by-case basis in the course of processing data items, or less frequently, to select a processing strategy for performing a given task (rarely do they help to decide what task to do next). In deciding what to do to data and how to do it, specifications received from users may prove insufficient, and certain system defaults may be inappropriate. Intelligent subsystems are needed to navigate such situations.

GIS users need help in whittling away options that aren't useful at given stages of their work, and could use advice on how to specify the options they decide to use. Attempts at providing such advice have been made using logic programming and expert system shells. Most applications of AI technology to GIS, however, attempt to automate data manipulations, such as deciding the size, format and placement of feature labels on maps (which may be the most popular testbed). This usually involves referring to a set of phenomena- and technique-based rules which are evaluated and weighed together to make tactical data processing decisions. Williams (1989) offers a good description how geographic (or any) expertise is cast into AI rules and tools:

> Intelligence can be achieved via two fundamental, but integrated sources. These sources are those of data relationships and structure, and techniques and procedures for manipulating and analyzing the data relationships. These sources can be considered as forming *expertise*. *Expertise* consists of knowledge about a particular domain (*real-world* geographic structures), understanding of the domain problems, and *skill* at solving some of these problems. Knowledge (in any speciality) is

> usually of two sorts: *public* and *private*. *Public knowledge* includes the published definition, facts, and theories of which textbooks and references in the domain of study are typically composed. But *expertise* usually involves more than just this *public knowledge*. Human experts generally possess *private knowledge* that has not found its way into the published literature. This *private knowledge* consists largely of rules of thumb that have come to be called *heuristics*. *Heuristics* enable the human expert to make educated guesses when necessary, to recognize promising approaches to problems, and to check effectively with errorful or incomplete data. Elucidating and reproducing such *knowledge* is the central task of building *expert systems*. (Williams, 1989, p. 558)

There is little difference between providing expert advice to users and applying this knowledge to automated procedures. Deciding what-to-do-next and how-to-do-it can be informed by consulting expert systems; it is a matter of style whether users choose to adjudicate this information themselves or let software handle such decisions. Capacities to exercise these options should be built into toolkits provided to users. While providing such capabilities may increase the possibility of "use error" (Beard, 1989), it can also help to avoid them by advising users of the nature and consequences of their assumptions, choices and actions.

The native intelligence of a GIS is highly conditioned by what information it maintains to qualify data items. The more such metadata about objects, locations and attributes that a GIS maintains, the more confidently it can be used. But how many GIS's document the level of encoding (Boolean, nominal, ordinal, interval, ratio, etc.) used for a given variable? How many store, much less can interpret, units of measure (e.g., persons per square mile, hectares, pH, ppm, BTUs, furlongs per fortnight) for items they maintain? Information can be lost, misinterpreted or made spurious unless appropriate operations are applied to data items, which requires that variables be qualified by their levels and units of measure, properties rarely included in GIS data models and definitions. As a result, it may be difficult for GIS users to switch between English and Metric units in thematic maps or reports they generate, or be warned that attributes created by adding or subtracting ordinal values may be meaningless. Users may not be sure what the units of a variable are, whether a given datum represents a percent, a category or a scalar; analytic functions may not "know" that multiplying income per capita with population density yields units of income density. Ignoring data quality information generates wrong results and incurs missed opportunities. In most cases, the mechanisms for handling properly qualified data are rudimentary; perhaps the fact that it is inconvenient for procedural languages to include metadata in function arguments has begged the problem, leading us to believe that dealing with it requires artificial rather than innate intelligence.

What might help GIS software deal with data quality better than it now does? Procedures need some additional intelligence to enforce metadata-based constraints, and will need to reference a few new data structures in order to do this. Implementations should avoid both transfer of large data objects between storage units and storage of seldom-used fields. OOP architecture may help in this regard: attributes of data objects such as parcels or rivers can be implemented as instances of classes of data, each of which has specific levels and units of measure, spatio-temporal and accuracy measures, and rules and methods for their manipulation. Because the same attribute class may be shared by otherwise unrelated data objects, any OOP environment which handles metadata in terms of classes must support multiple inheritance (non-hierarchical object composition, as when describing the wetlands contained in a lot or the parcels occupying a swamp).

**Summary and Conclusions**

Geographic Information Systems have been a commercial commodity for nearly a decade. This technology has proven its value in land record systems, environmental impact analysis, facility management, land development, urban planning and other areas of application. Trained GIS users are in short supply, partly because so many of their skills are improvised and so much of the knowledge required to run GIS applications is undocumented. But as GIS databases grow larger and the features they encode become more complex, the pathways of spatial analysis tend to multiply and results become more equivocal. Unless we can bring more intelligence to bear, our GIS applications may not continue to yield as useful, interpretable and replicatable findings as we might wish.

To effectively apply geographic information systems, users need better tools for constructing, navigating and processing databases. Vendors cannot hope to provide users with "solutions" to very many geoprocessing problems in the form of full-featured and fully-documented applications. Rather, they should concentrate on adopting data models that better express spatial structure and process, software tools that extend both system functionality and ways to apply it, and advisory subsystems that can identify semantic subtleties when users query and analyze spatial-temporal data. GIS architecture will have to change at micro, meso and macro levels to enable these capabilities. New paradigms, unfamiliar data structures and relatively unproven techniques may have to be deployed. In the process, researchers, developers and users will have to alter their modes of operation. Geographic

Information Systems have already changed the ways in which we think about maps; the time has come to change how we think about, build and use this potent but imperfect technology.

## References


Beard, Kate (1989). *Use error: The neglected error component.* **Proc. Autocarto 9**. Falls Church, VA: ACSM, pp. 808-817.

Berry, Brian (1964). *Approaches to regional analysis: a synthesis.* **Annals**, Assoc. of American Geographers, v. 54, pp 2-11.

Chrisman, Nicholas (1984). *The role of quality information in the long-term functioning of a geographic information system.* **Cartographica**, vol. 21, nos. 2&3, pp. 79-87.

Cooke, Donald (1989). *TIGER and the "Post-GIS" ERA.* **GIS World**, vol. 2 no. 4, ps. 40 *ff*.

Cooke, Donald and William Maxfield (1967). *The development of a geographic base file and its uses for mapping.* **Proc. URISA**.

Corbett, James (1977). *Topological principles in cartography.* **Proc. Autocarto II**. Falls Church, VA: ACSM, pp. 61-65.

Cox, Brad (1987). **Object Oriented Programming: An Evolutionary Approach.** Reading, MA: Addison-Wesley, 274 p.

Dutton, Geoffrey (1984). *Geodesic modelling of planetary relief.* **Cartographica**. vol. 21. nos. 2 & 3, pp. 188-207.

Dutton, Geoffrey (1989a). *Planetary modelling via hierarchical tessellation.* **Proc. Autocarto 9**. Falls Church, VA: ACSM, pp. 462-471.

Dutton, Geoffrey (1989b). *The Fallacy of Coordinates.* **Multiple Representations: Scientific report for the specialist meeting.** B. P. Buttenfield and J. S. DeLotto, eds. National Center for Geographic Information and Analysis, tech. paper 89-3, pp. 44-48.

Fekete, Gyorgy (1990). *Rendering and managing spherical data with Sphere Quadtrees.* **Proc. Visualization 90**. New York: ACM.

Goodchild, M. (1988). *Stepping over the line: Technical constraints and the new cartography.* **The American Cartographer**, vol. 15, no. 3, pp. 311-319.

Goodchild, Michael and Yang Shirin (1989). *A hierarchical data structure for global geographic information systems.* Santa Barbara, CA: National Center for Geographic Information and Analysis, tech. paper 89-5.

Kacmar, Charles (1989). *A process-oriented extensible hypertext architecture.* **ACM SIGCHI Bulletin**, vol. 21 no. 1, pp. 98-101.

Langran, Gail (1989). *Accessing spatiotemporal data in a temporal GIS.* **Proc. Autocarto 9**. Falls Church, VA: ACSM, pp. 191-198.

Peucker, Thomas and Nicholas Chrisman (1975). *Cartographic Data Structures.* **The American Cartographer**. vol. 1, pp. 55-69.

Peuquet, Donna (1988). *Issues involved in selecting appropriate data models for Global databases.* **Building Databases for Global Science**. H. Mounsey and R. Tomlinson (eds.). London: Taylor & Francis, pp. 187-260.

Sabella, P. and I. Carlbom, (1989). *An object-oriented approach to the solid modeling of empirical data.* **IEEE Computer Graphics & Applications**. vol.9, no. 5, pp. 24-35.

Tomlinson, Roger (1988). *The impact of the transition from analogue to digital cartographic representation.* **The American Cartographer**. vol. 15, no. 3, pp. 249-261.

Upson, Craig, Thomas Faulhaber, David Kamins, David Laidlaw, David Schlegel, Jeffery Vroom, Robert Gurwitz, Andries van Dam (1989). *The Application Visualization System: A computational environment for scientific visualization.* **IEEE Computer Graphics and Applications**. vol. 9, no. 4, pp. 30-42.

White, Denis and Jonathan Corson-Rikert (1987). *"WYSIWYG" map digitizing: Real time geometric and topological encoding.* **Proc. Autocarto 8**. Falls Church, VA: ACSM, pp. 739-743.

Williams, Robt. (1989). *Geographic Information: Aspects of phenomenology and cognition.* **Proc. Autocarto 9**. Falls Church, VA: ACSM, pp. 557-566.