GIS FUTURE: AUTOMATED CARTOGRAPHY OR GEO-RELATIONAL SOLID MODELLING?

Hrvoje Lukatela
2320 Uxbridge Drive
Calgary, AB - T2N 3Z6 CANADA
(Envoy 100: lukatela)

ABSTRACT

Computer Mapping and Geographic Information Systems have evolved into two distinct computer application areas. While the two share some common ground, their differences are significant enough to merit an independent search for the basic software design strategy.

This paper concentrates on elements which characterize and define Geographic Information Systems - in contrast to Computer Mapping. It then explores their influence on system design criteria and software engineering aspects, such as the data modelling, spatial operators, external storage management, reference surface selection and computational geometry.

PURPOSE

Due primarily to the reasons of history and technical tradition, Geographic Information Systems are routinely built using the software design fundamentals originating from the much older field of Computer Mapping. The purpose of this paper is to contribute to the discussion of alternatives, which could result in better designed software and higher functionality of such systems.

Issues are examined strictly from the technical perspective of a system designer: operational value of the system will, in addition, depend on the organizational issues, which can be analyzed according to similar criteria.

DEFINITIONS

Computer Mapping Systems (CMS) are applications which store and manipulate location-defining attributes of data objects, with the purpose of generating their analog, graphical representations. Those can be either permanent or transient, and can emphasize spatial relationships between objects ("overlays" etc.) or include graphical portrayal of additional, non-spatial data attributes in spatial context ("thematic cartography"); but their utility is restricted to the visual consumption by a human operator or system user.

Geographic Information Systems (GIS) are applications based on a data model dominated by the location-defining attributes of its objects, capable of data processing required by an administrative, technical, educational or other discipline, in order to automate some of its functions and processes. While the generation of analog view of data can be (and almost always is!) included in the functional repertoire of the system, it is not it's primary function. Indeed, GIS often use a combination of location-defining and other attributes in processes which mimic application-domain inferences, procedures or depictions, and produce results which are output not in cartographic form, but in the form native to the application itself (e.g. report, table, decision selection, statistical graph or control loop signal).

In short, Computer Mapping Systems automate the process of composition and production of analog map documents; Geographic Information Systems use "digital maps" in order to perform functions intrinsic not to the cartography, but to some practical discipline that studies geographically distributed data.

DATA MODELLING AND SPATIAL OBJECT CLASSES

CMS data models are usually based directly on the graphical scheme that is at the same time a precise description of the system output. It is typically restricted to 0 and 1 dimensional objects, (points and lines), and includes attributes which specify the details of graphical presentation (symbols, colour, line style, label placement, etc.). The system can also include non-spatial attributes, or be partitioned into pre-defined "layers", representing different classes of spatial objects.

Since GIS are primarily application systems serving an unending array of industries and disciplines, "GIS data model" can not be addressed in a generic form. Generalized, application-independent theory of data modelling enjoys currently the research attention beyond anything that a specialized field like a GIS can convoke. Results are directly applicable to GIS objects which are not spatially-defined, and also to the non-spatial attributes of spatially defined objects.

The geometry of spatially defined real-life objects will usually be significantly more complex than the geometry of a CMS graphical scheme. This will be caused - typically - by a combination of more complex geometry, and by the temporal nature of the object shape, size or location. It is in this area that GIS requires specialized modelling techniques.

Invariably, practical system design considerations will require some simplification as a part of the process of transformation from the object into its digital representation. The central problem of the spatial data modelling consists therefore of establishing the balance between the simplicity and faithfulness of the spatial object representation: overly complex representation will make the system to costly to construct and operate; insufficient faithfulness will impair the functional power of the system and thus reduce the benefits of its implementation.

While many CMS data models are built using only simple point and line objects, GIS are usually required to model objects with more complex spatial or spatially-temporal definition. Those listed below - in the order of increased complexity - probably represent the most common classes of spatial objects stored on GIS:

1) **Point set**: a finite number of surface point locations. (The set is aggregate, and all non-spatial attributes pertain equally to all locations in the set.)

2) **Discrete surface value set**: point set with a single, numerical value associated with each location in the set.

3) **Line set:** one or more ordered point sets.

4) **Gravitational movement trace:** parameters defining geometry of the conic section, its external orientation parameters in the global frame of reference, and a point on the curve at the epoch.

5) **Kinematic movement trace:** ordered series of surface point locations (possibly including normal displacement coordinate), with time value associated with each.

6) **Surface region:** one or more ordered point sets, representing boundary rings of a non-simply connected surface region. (Boundary must not cross itself, and ring directions must be consistent among all the rings in the set.)

7) **Boundary system:** a composite object consisting of an ordered list of single-location point sets representing the node points and an ordered set of line sets representing the boundary segments. (A series of boundary-system-object/node--point/segment identifiers can be used as an alternate spatial definition of a single-ring region object.)

GIS spatial object lists, such as the one above, are by nature open-ended. A complete list can only be composed based upon a valid data model for a specific application. The designer of application--independent GIS software tool must, however, take a more pragmatic approach: a finite collection of objects must be selected and implemented as intrinsic to the package. The application builder can then restrict his data model to the objects supported by the tool, or extend it by providing additional data structures and functions in the application software.

Either the tool - if it is used - or the application code must provide a complete set of spatial operators, capable of deriving spatial unions and intersections between all union-compatible pairs of object classes. In CMS systems such operators are used on their own, and the results of their invocation are displayed graphically. In GIS, spatial operators are frequently combined with processing based on non-spatial attributes in complex, response-time critical transactions, which do not tolerate relatively low level of efficiency of spatial operators commonly found in CMS. In addition, such transactions can create new spatially-defined objects, which the system must be able to treat in exactly the same way as source objects - e.g. display, store on the data base, export to another system etc.

Different objects will in general be spatially defined with different levels of precision, and their representation should take this into account by using different internal coordinate item width. Since this can be provided only in discrete steps (e.g. single and double precision floating point representation of coordinates) each object must carry an item which indicates, numerically, spatial precision (as opposed to the artificial data resolution) with which the object is known to the system. It is worth noting that the absence of such indicator will influence CMS only in a limited manner: once scaled down to the size of its graphical depiction, precision related problems will be insignificant compared to the same in GIS, where spatial relationships are evaluated in object-space size and precision.

The level of spatial precision required for various data objects depends on the target system output precision, but also on the data model characteristics:

As long as the performance is not critically affected, geometry elements that can be derived from the primitive location attributes should not be stored on the system, but derived as, and when, required. However, primitive attribute precision requirements can often be relaxed, if the precision-critical elements are stored redundantly. (Common examples include precise land-parcel areas, or distances along the centre-line of a liner feature, stored explicitly, in a combination with point coordinates scaled from the map.) Such inconsistencies in the geometry model can severely restrict implementation of functions which have not been "built-into" the original design; principles of "open-ended" system design influence equally the spatial and non-spatial elements of the data model.

EXTERNAL STORAGE MANAGEMENT

One of the common characteristics of CMS and GIS is the high data--volume brought about primarily by the nature of location defining attributes. The data retrieval patterns, however, are different. In CMS, most partial retrievals will be geographical, limited to the current location of the display window. Special-purpose storage indexing schemes (based mostly on various algorithmic representations of regular planar tessellation - e.g. B-trees, Quad-trees, R-trees etc.) have been both well researched, and verified in practice in many generic CMS packages. Variety of objects stored on a CMS data base is usually restricted to a relatively small number of fixed "layers" or "record types". Comparatively low volatility of CMS data bases makes system implementors and operators relatively undemanding in the areas such as on-line update transaction support, access control, ease of backup/restore process, encryption, multiple update conflict resolution, and a large number of other facilities considered sine qua non in a modern data base software package.

GIS data bases parallel those of most large data base development projects, but must, additionally, allow spatially-defined retrievals.

Current preponderance of the relational data base model has significant repercussions on the whole GIS realm. Both the application-domain expert and the application programmer are likely to demand and expect the flexibility and conceptual simplicity associated with the relational model on both the non-spatial and spatial data base development projects. From their viewpoint - when it comes to manipulations performed by the data base manager software - there must be little or no difference between the spatial and non-spatial attributes of their objects.

If that is the case, spatial retrieval schemes and attribute storage must follow the general philosophy of the relational data base model in several important aspects:

- Spatial relationships must not be encoded with the data (in form of pointers, "topology indicators", spatial structure descriptors, etc.), but must be derived from the location--defining data attributes at the time of retrieval processing.

- Spatial retrievals must be possible based not only on the pre-defined surface elements, but also on relational algebra productions between objects on the data base.

- Introduction of spatial criterion into a complex retrieval selection set must not complicate the retrieval request formulation more than would be the case if an additional non-spatial criterion was introduced.

- Redundancy criteria and degree of normalization of spatial attributes must equal those applied to non-spatial attributes.

GIS data compilation that does not violate above principles can be called **"geo-relational"** data base. Both the application programmer and end-user alike will perceive it as a relational data base in which spatial and non-spatial attributes have been integrated in a seamless manner.

Since few projects can justify dedicated efforts required to implement complete data base manager software, GIS system designer has two alternatives: a generic CMS package which provides some degree of non-spatial attribute support, or a general-purpose data base package with the addition of functions providing spatial data storage and retrieval. It appears that at present the former alternative enjoys greater popularity among GIS system developers. This might change: emphasis on the relational model and associated flexibility of the data base design, combined with the increased demands on the operational strength of the data base, makes current generation of data base products very attractive to the GIS implementor. This might, in turn, provoke the emergence of software products which will provide the necessary set of geo-relational functions in the form of specific data base manager "add-on" packages.

REFERENCE SURFACE AND COMPUTATIONAL GEOMETRY

Cartographic projection plane is the spatial data domain of most CMS. This is not the case with GIS: even when coordinates used as location defining attributes are e.g. Universal Transverse Mercator (UTM) projection plane "northings" and "eastings", they conceptually represent locations on the surface of the Earth. This becomes obvious when the need arises to model an object which extends across two different UTM projection planes: the object itself (unlike parts of its depiction) does not belong to two distinct data domains! In GIS, cartographic transformations - if used at all - only "mask" the spatial data domain by defining it implicitly by the way of their own parameters and algorithms.

True spatial data domain of GIS is always part (or whole) of the planetary (or "reference") surface, or, in different words - **reference surface is the dominant spatial object of every GIS.**

In order to formulate computational geometry - a set of rules used as a base for derivation of spatial relationships between the objects - the reference surface must be defined in a simple, yet sufficiently precise form. Plane, sphere and two-axial ellipsoid are commonly used for this purpose. (Since the reference surface interacts with every other spatial object, more complex reference surface definitions are of little or no practical value to a GIS designer.) The simplicity-

-precision scale is obvious: plane is the simplest, and ellipsoid the most precise GIS reference surface.

Before the question of "sufficient precision level" is examined, it is important to note that - by definition - the computational geometry used (planar, spherical, or ellipsoidal) must yield all numerical results with the level of precision required by the system, without abandoning the metrics implied by its definition. As an example, if the system uses UTM coordinates to define locations, but "corrects" coordinate diagonal (by applying UTM "scale factor") to obtain distances between two points, its computational geometry (and therefore the reference surface) are clearly not planar, but ellipsoidal, defined implicitly by the correspondence of UTM and ellipsoid coordinates.

As mentioned before, all objects modelled by the system will not be known with the same spatial precision - or represented numerically with the same resolution. The inaccuracies introduced by the spatial frame of reference must not lower the accuracy of the most precise derived data item generated by the system. In general, this will be achieved if the distortions introduced by the geometry of the reference surface are kept at least a full order of magnitude below the resolution of highest-precision data items.

Few - if any - GIS can be constructed using the plane as the reference surface, without violating this principle. (As an example, a simple municipal cadastral and engineering data base, extending over an area with a radius of only 20 Km, with coordinates defined to the precision of .1 m can not be constructed in plane, if the geometry of objects larger than some 250 meters in diameter is to be generated from the coordinate data!) Planar cartographic projections are therefore of little value as GIS reference surfaces; their use should be restricted to the necessary conversion of coordinate data on output and input from and into GIS.

Spherical models - particularly those based on spherical coordinates obtained by either rigid or approximate orthomorphic transformation from the ellipsoid - are much more likely to satisfy GIS reference surface precision requirements. Using the same municipal system example as above, the radius of the area of coverage would have to extend more than ten-fold before the same distortion is encountered. Another example: single-plane data domain GIS covering the contiguous United States are commonly constructed using a "two-parallel" Lambert projections. Maximum (local!) linear distortion of such system will be as high as 1 in 50. Local linear distortion of the same system based on a single orthomorphic sphere would be only 1 in 1500 - accuracy level almost approaching that of a single UTM plane system (1 in 2500). In addition, spherical computational geometry is based on simple, closed numerical relationships, which, compared to planar calculations, require no more than a modest (typically in the order of 50%) linear increase in computing power.

Large-area GIS, particularly those that include numerical data representing distance or direction measurements carried out at object-scale, may require ellipsoid reference surface in order to attain the required spatial precision level. The ellipsoidal computational geometry presents the algorithm designer with a significantly more complex series of problems, and may require dramatic increase in computing power.

While such calculations are still practical in case of low data--volume problem solutions, high data-volume problems - as, for instance, evaluation of spatial unions and intersections - will require better techniques than those which are considered satisfactory for planar or spherical systems.

One class of GIS makes use of ellipsoidal spatial reference system almost mandatory: systems that relate data acquired from orbiting sensors and terrestrial data originating from a large variety of conventional sources. (Present stratagem of pre-casting the sensor data into some particular projection plane geometry, scale and pixel--ratio is satisfactory as a base of CMS, but of little or no usefulness for GIS.) Manipulation of high-density sensor data will require techniques similar to those necessary for the evaluation of spatial unions and intersections. Once developed, such techniques will make possible the solution of complex, high data-volume spatial/terrestrial problems in the most logical frame of reference for their solution: one based directly on the metrics generated by the same force that shapes the terrestrial surface, and determines the trajectory of a free-falling sensor: the inverse r**2 force!

CONCLUSION

The primary design problem of most Geographic Information Systems is that of integration of two classes of data and procedures: those that define spatial characteristics of objects, and those that describe complex, application-specific qualities and measures of same objects.

Digital modelling of spatial objects in GIS must be optimized toward object-space precise and efficient evaluation of complex relationships defined by a combination of spatial and non-spatial criteria; numerical models developed for the purpose of either manual or computerized map production are not adequate for this purpose.