GIS AND UC BERKELEY'S MULTI-VALENT DOCUMENT ARCHITECTURE

Howard Foster, Programmer,
Computer Science Division, University of California Berkeley

Abstract

Some unique features of the UC Berkeley Digital Library Project influence the application of GIS technology. These are (1) a significant number of documents relate to the California natural environment; (2) the Project must accommodate environmental planners who often need to make complex ad-hoc queries on multi-attribute data; (3) the Project almost totally relies upon the World Wide Web for document access; and (4) Project documents are organized as "Multivalent" (multi-layered) entities for which GIS operations must be integrated. In response to these requirements our GIS has a prominent database component to accommodate complex queries on spatial and non-spatial attributes; our GIS depends upon a distributed arrangement of components, especially a Java image for adequate display functionality (e.g., pan and zoom). Ultimately our GIS is to be integrated into the general multivalent document access interface. This means, for example, a project site map accessed within a document could turn into a GIS interface for the analysis of a project's contextual geographic information.

The Berkeley Digital Library Project's Multi-Valent Document architecture

Berkeley's Digital Library Project [Wilensky] is developing new storage, retrieval and interface technologies for heterogeneous documents, for access from the World Wide Web. Our particular corpus, California environmental documents, is particularly diverse and includes traditional (paper) reports such as EIRs and bulletins and also non-traditional library documents such as ground photography, aerial photographs, satellite imagery, time series data, geographic coverages, etc. One of our innovations is the invention of a Multi-Valent Document (MVD) architecture [Phelps] which is intended to help developers and end-users add functionality to legacy documents (documents for which there is an established representation of the original work). In the MVD architecture, document description and access methods are implemented as layers plus methods. For example, an image of an original page might be supplemented by a plain-text layer, derived by OCR, which can be searched for keywords, or a hyperlink layer which connects to commentaries, annotations, and bibliographic references, or a mathematical layer linking document equations to formulae in a programming language such as Mathematica®. (It can be noted that the MVD design already has a GIS feel because it uses spatial information—the pixel coordinates of text and other document objects—to link together layer information. GIS technology within the MVD scheme is to provide linkages to contextual geographic information

In the MVD world it can be imaged that as a reader leafs through the digital form of a legacy document, such as an EIR, the reader encounters a map of a project's boundaries. If the reader is an environmental planner, contextual questions might arise as to how much of the project site is composed of designated wetlands or how much of the project site is within a FEMA-designated floodplain. These are typical GIS queries which require traditional GIS functionality, such as the listing of relevant coverage information, selection and display of vector and raster information, and pan and zoom. In the MVD world, a map is a gateway to contextual environmental information identified by the geographic coordinates of the map image.

Design requirements for Berkeley Digital Library GIS technology

A number of factors are shaping the design of Berkeley's GIS efforts. These are: (1) our GIS technology must integrate well with other MVD document representations; (2) in our view, typical environmental planning queries involve relatively simple spatial criteria (i.e., satisfied by co-incidence tabulation and distance buffering) but often require sophisticated manipulation of non-spatial attributes (e.g., "show only commercial and industrial areas", "show the best agriculture suited soils"); (3) our GIS technology should provide a coherent user interface which clearly indicates selection and rendition choices; (4) we need to provide for site, vector, and raster data and we need to provide for layer overlay to accommodate complex queries; (5) our technology must be extensible to allow us the use of the processing capabilities of other technologies.

In addition we believe that we should provide for the following capabilities:

1. Region setting by (a) access to named extents (e.g., Sonoma County), (b) pan and zoom
2. Display a list of coverages intersecting the current region
3. Pan and zoom
4. Vector overlay

5. Raster overlay by (a) Null or 0-value transparency, (b) transparent colors
6. Masking
7. Surface query (query of the attributes of a selected layer at the mouse location)
8. List display of attribute categories and values
9. User selection of display attributes—colors and color schemes
10. Area tabulation of multi-layer attribute combinations
11. For advanced operations (e.g., modeling), a gateway to layer creation outside of the browser for established operations, e.g., buffer creation, map algebra

Our design efforts are also guided by the following assertions about the current state of GIS technology and the World Wide Web

1. Although there are many promising starts, there is currently no satisfactory Web GIS. We shouldn't wait for the major GIS vendor to produce such a product as it may not fit our needs (see above).

2. Current Web GISs provide only browser level functionality (i.e., the display of complete coverages as map images); we need to provide for data selection within coverages.*

3. Web GIS technology is unfolding in three phases. (1) Past—complete server-side interpretation of GIS queries which result in the creation of gif images which are displayed by a dumb client. (2) Present—enhanced client capabilities to process site, vector, raster, and tabular (attribute) information. (3) Future—GIS clients are part of a Web-object infrastructure where clients access data-method objects, many of which are not local.

4. It is desirable to use database technology to the greatest extent possible. Database technology supports many geographic-relevant functions (e.g., point-in-polygon) with high performance suited to the web. Database technology is completely sufficient for simple geographic information such the display of site data and the retrieval of geo-located objects (e.g., air photos). When specialized spatial attribute processing is required, it can be provided by external functions.

5. Since we desire real-time overlay capabilities, area features are to be represented as multi-attribute raster data.

6. Our efforts are to be incremental, and non-proprietary technologies are to be used to the greatest extent possible.

In an early design cycle we devised seven queries, which we believe to be representative of those asked by our environmental planning clients.

1. Display the soil types known to contain rare plants.
2. Display only the soil areas that have rare plants on them.
3. Display landuse aggregated by landuse type.
4. Display landuse types as aggregated by user-specified criteria.
5. Display industrial and commercial landuses within 4 meters of sea level, on a satellite image.
6. Display the three most common soil types of Marin County.
7. Display all soils whose porosity is greater than 'Cohassat clay loam'.

Putting it together—what the user will see

At its core, the Berkeley Digital Library Project GIS is true to the basic MVD idea of manipulation of layers and behaviors. A user (first) selects a geographic layer, and (second) manipulates that layer's graphical and logical behavior. Layer selection follows the traditional list selection metaphor; a layer is selected from a list of layers whose bounds intersect the current region. If the data are multi-attributed, then a sublist of map attributes is presented and the user will select from this list. If the user wishes, he/she can further define subselection criteria (e.g., slopes over 30%), or define arbitrary selections by placing checks next to data values. The selection information can be turned into (geo-spatial) SQL queries (see later in this paper). Once layer information is defined, display rendition options can be specified such as "default", "red", "greyscale" or "USGS Anderson landuse colors". Simple, single layer queries might end at this point with, perhaps, the user panning and zooming to different areas.

* One exception is GRASSlinks [Huse] which provides for attribute selection with its reclass facility. Attributes mapped to positive values are displayable, unmapped attributes assume null values. Reclass layers have full GRASSlinks layer capabilities—they can be displayed, act as masks and participate in crosstabulations.

To view more complex environmental information, the user will need to consider multiple layers of information. (e.g., the land uses within 500 meters of the Rogers Creek Fault, or Marin County wetlands). In this case the user is also presented with a two step process: (1) pick a layer, as in the above example, and (2) define its graphical and analytical behavior relative to other layers. Multiple layers of linear features and boundaries, represented as vectors, are comparatively easy to display as these are typically sparse and lines can be opaque without too much hiding of the information below. Layers composed of area coverages are more challenging. The current options under consideration are (1) Null and/or 0-value transparency, (2) transparent colors, (3) masking. Null-value transparency is easy to consider. The user should be able to specify layer areas with non-null values to be opaque while null values are to be transparent to the layer below (in null-value areas, the underlying information shows through). If a user makes a selection of e.g., peat soils of Sacramento County, and overlays these on a satellite image, the user will see the land-cover context for such soils. Transparent colors provide a more sophisticated means to view relationships between layers. For example, the user might want to overlay a soils map (e.g., rainbow gradient colored by soil depth) on top of a shaded relief map (greyscaled). By tinting the greyscale image with the soil colors, the user can see the relationship of soil depth to landform factors such as slope steepness and orientation. Transparent colors might not be suited to layers that are all rendered as colored images because blended colors might be difficult to interpret, but color schemes are under user control and can be manipulated appropriately. Masking is the third option we would like to offer for the visualization of multi-layer information. Masking has both graphical and analytical consequences and mask semantics need to be clearly communicated to the user. Our approach is as follows: (1) Make a selection for a layer as described above; (2) specify that the layer is to act as a mask with either (a) non-null (valued) areas are to be transparent and null layers block, or, alternatively, (b) valued areas are to block and nulls are transparent; (3) specify masking depth, where masking applies to (a) the immediately adjacent layer only, or (b) to all layers. For example, expressed as a function, a mask might be expressed as in a geo-spatial SQL as MASK((select height from elev where height < 4), value-transparent, above-layer-only). Sophisticated queries could utilize a number of mask elements.

One final capability must be communicated to the user--this is the ability to change layer assignment order. This is a user interface issue with an "up/down" function indicated in the margin tab of each layer.

For example, for query #5—"Display industrial and commercial landuses within 12 feet of sea level, on a SPOT® (satellite image) map" the following procedures would be invoked:

A: Define the current region.

B: Select the landuse layer attributes to be displayed from a list; define the color scheme as "red", and specify 0-value transparency.

C: Select elevation layer with its height values less than or equal to 4 meters; specify this layer as a mask, to apply only to the above layer, and mask (make null) parts of the above image not meeting this selection criteria.

D: Specify the underlay image; in this case a SPOT satellite image.

The combined geo-spatial SQL might look something like:

top layer:
display((select type from genplan where label like '*industrial*' or label like '*commercial*'), red, 0-value-transparent);

middle layer:
mask((select height from elev where height < 4), above-only, null-values-blocked);

bottom layer:
display((select val from spot), grey-scale);

Getting from here to there

The Berkeley Digital Library Project's GIS browser is composed of three processing modules: (1) a JAVA client, (2) a RDBM server to process attribute data and (3) a spatial data server to provide spatial services such as buffering and overlay reporting. Our design approach is evolutionary in which most operations start out on the server side; as the project evolves more and more functions will migrate to the client. The JAVA client is being written from scratch by Loretta Willis and server operations, accomplished with scripts calling GRASS and Illustra functions, are being written by Fah-Chun Cheong and Howard Foster.

Why GRASS and Illustra®?

Illustra (now being merged into the Informix® line of products) is an extremely flexible OO-RDBMS which can utilize external functions. External functions are often packaged as "datablades" and, in this case, many GIS-relevant functions (e.g., point-in-polygon and polygon-in-polygon) are provided by the 2D datablade and such queries can be made more efficient by R-Tree indexing. Illustra also provides for graph-storage of vector data, thus "shortest distance" queries can be answered. One disadvantage of Illustra is that it is a proprietary and perhaps expensive. We would like to use Illustra only for rapid prototyping. Postgres95, a publicly-available database manager, shows promise for GIS data processing as it also has the ability to utilize external functions and *-tree indexing.

GRASS was developed by the US Army Corps of Engineers in the mid-1980s and is a proven raster GIS with open source code which allows selected functions to be ported to Illustra or Postgres95. Also, ironically, while GRASS itself has no well-developed interface for a DBMS,[*] it is exceptionally database friendly. Important attributes of GRASS data are defined by ASCII tables (e.g., location, colors, categories, text labels, etc) and these can be externally manipulated by the database manager. Also GRASS provides for virtual layers, defined by tables of category mappings, and these too can be manipulated by the database manager. (It is by the processing of these tables that multiple layer attributes can be displayed.) Also, GRASS's data formats are close to generic GIS data interchage formats such as ARC/INFO's® ungenerate format.

The role of geo-spatial SQL

To evaluate GIS functionality for our clients, it has been very useful to consider GIS operations as SQL queries (the geo-spatial SQL answers to the seven queries has identified needed processing capabilities). But the project roll of geo-spatial SQL has not yet been determined. Because of our evolutionary approach, GIS implementation will involve JAVA methods, CGI perl scripts and traditional SQL queries at least quite a while and we do not yet know of the advantages and disadvantages that a single geoquery-language/parser/module approach will give us.

Current progress

Our GIS research has been extremely useful and we already have implemented database query and retrieval of geographic site data and geographic objects. We currently have a JAVA prototype which provides for panning and zooming of multi-resolution image data (San Francisco Bay Area USGS Digital Orthophoto Quads) and for the overlay of USGS Digital Line Graph road data. Server side technologies are being developed in parallel. We hope and believe that we will provide useful GIS technology to our environmental planning clients and to take advantage of the opportunities brought about by the extremely rapid pace of World Wide Web technology development.

References

Farley, J.A., and Song, W., Attribute Management, Data Integration, and GIS Links to SQL Databases. Geographic Information Systems: Proceedings of the Seventh Annual GRASS Users Conference August 1993, pp. 207-218. Gary Waggoner, ed., National Park Service Natural Resources Publication Office.

Huse, S. GRASSLinks: a New Model for Spatial Access in Environmental Planning. Ph.D., dissertation, University of California Berkeley 1995. Available from http://www.regis.berkeley.edu/sue/phd.

Phelps, T. A., and Wilensky, R. Toward Active, Extensible, Networked Documents: Multivalent Architecture and Publications. Proceedings of the 1st ACM International Conference on Digital Libraries, Bethesda, Maryland, March 20-23, 1996, pp. 100-108. Available from http://elib.cs.berkeley.edu/papers.html).

Wilensky, Toward Work-Centered Digital Information Services. Available from http://elib.cs.berkeley.edu/papers.html.

[*] There have been a number of database modules and API developments for GRASS but none have been widely distributed. See e.g., [Farley].