Hydrographic Data Management using GIS Technologies

Neal G. Millett and Simon Evans

Environmental Systems Research Institute, Inc., 380 New York St.,
Redlands, CA 92373-8100

I. Abstract

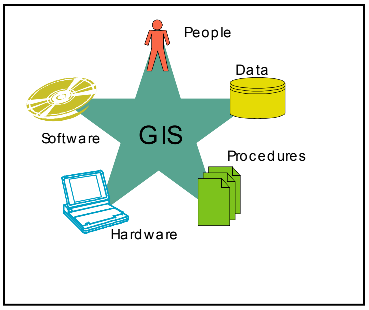
Recent advances in marine data sensing technologies, such as multibeam echo sounders, LIDAR, and high speed, high resolution side scan sonars raises a series of data management challenges for the hydrographer. The days of small, single point data collection are vanishing fast and it is now more appropriate to ask how many gigabytes of data will be generated daily or even hourly during survey operations. With vast amounts of data now a reality, the questions of data storage, maintenance and access must be addressed. Additionally, marine data collection has become extremely expensive. It would seem logical that there is an emerging need for efficient and precise data management systems to store, maintain, and distribute extremely large data sets.

This paper will promote the use of Geographic Information System (GIS) technology in order to provide effective and efficient data management for processed hydrographic data. The use of GIS in the marine industry has been restricted in the past due to hardware and software limitations. Environmental Systems Research Institute, Inc. (ESRI), has recently completed research on an implementation of an object-oriented GIS data model (Geodatabase) and applications (ArcGIS). The model lends itself well to database designs and applications associated with the collection, documentation, distribution, and analysis of large amounts of vector, raster, and surface modeling data (i.e., a data warehouse). It also provides the end user with the options of setting states and behaviors for each data object. Using this approach provides the means of capturing/modeling ocean data, enabling the existence of "intelligent data objects" and methods to distribute the data to end-users.

II. Background

What is GIS?

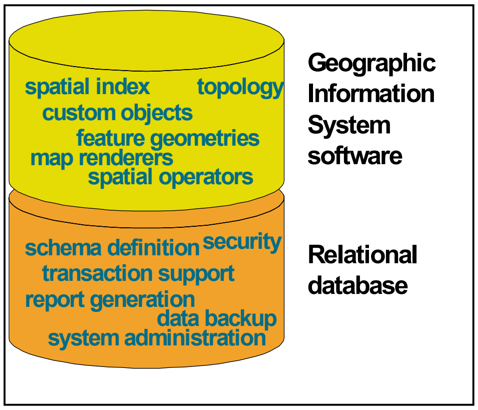
A geographic information system is the combination of skilled persons, spatial and descriptive data, analytic methods, and computer software and hardware—all organized to automate, manage, and deliver information through geographic presentation [1].



A GIS processes any data that has a spatial component. For hydrographic purposes, this information is quite diverse—it can be aerial photographs or satellite imagery, high resolution sonar data, multi-beam soundings, or LIDAR data.

GIS software: a Geographic Database

The key concept to grasp about GIS software is that it is, in fact, a geographic database management system. Geodatabases are implemented directly on commercial relational or object-relational database management systems. The reason for this is to leverage the capabilities of commercial database software, which include data backup, table definition, transaction management, and system administration tools. A GIS extends an object-relational database so that it can efficiently store geographic data, produce maps, and perform spatial analytic tasks. ArcGIS currently can support geodatabases implemented on Oracle 8, SQL Server, Informix, DB2, and Sybase.



Some of the functions that GIS software adds to an object-relational database management system are:

- The ability to store the geometric shapes of features directly in a database column.
- A framework to define map layers on data and specify drawing methods; these can be drawn based on attribute values.
- An infrastructure to support the creation of simple and sophisticated maps. Many common map-making tasks are simplified.
- The creation and storage of topologic relationships that exist among features, such as network connectivity and integrated polygon topology.
- A spatial index spanning two dimensions for rapid retrieval of geographic features.
- A set of operators for determining geographic relationships such as proximity, adjacency, overlay, and spatial comparison.
- Many tools to support spatial queries such as network tracing and polygon overlay analysis.
- A work-flow system that allows the editing of geographic data by many users and manages versions.

A GIS is thus a spatially enabled database management system. This architecture gives you the best of commercial database technology and sophisticated geographic software.

III. Data Representation Models

With a GIS, data can be modeled in three basic ways: as a collection of discrete features in vector format, as a grid of cells with spectral or attribute data, or as a set of triangulated points modeling a surface [1].

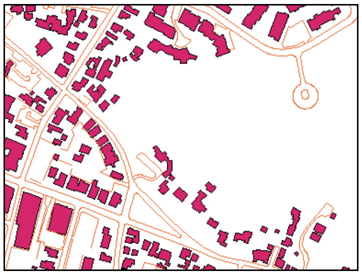


Figure 1 Vector Data

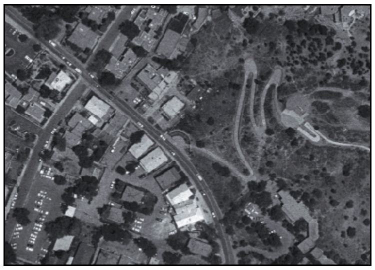


Figure 2 Raster Data

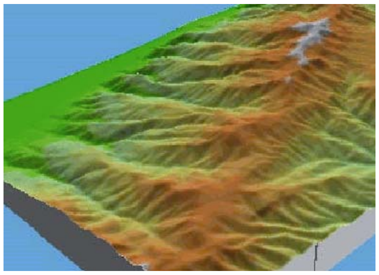
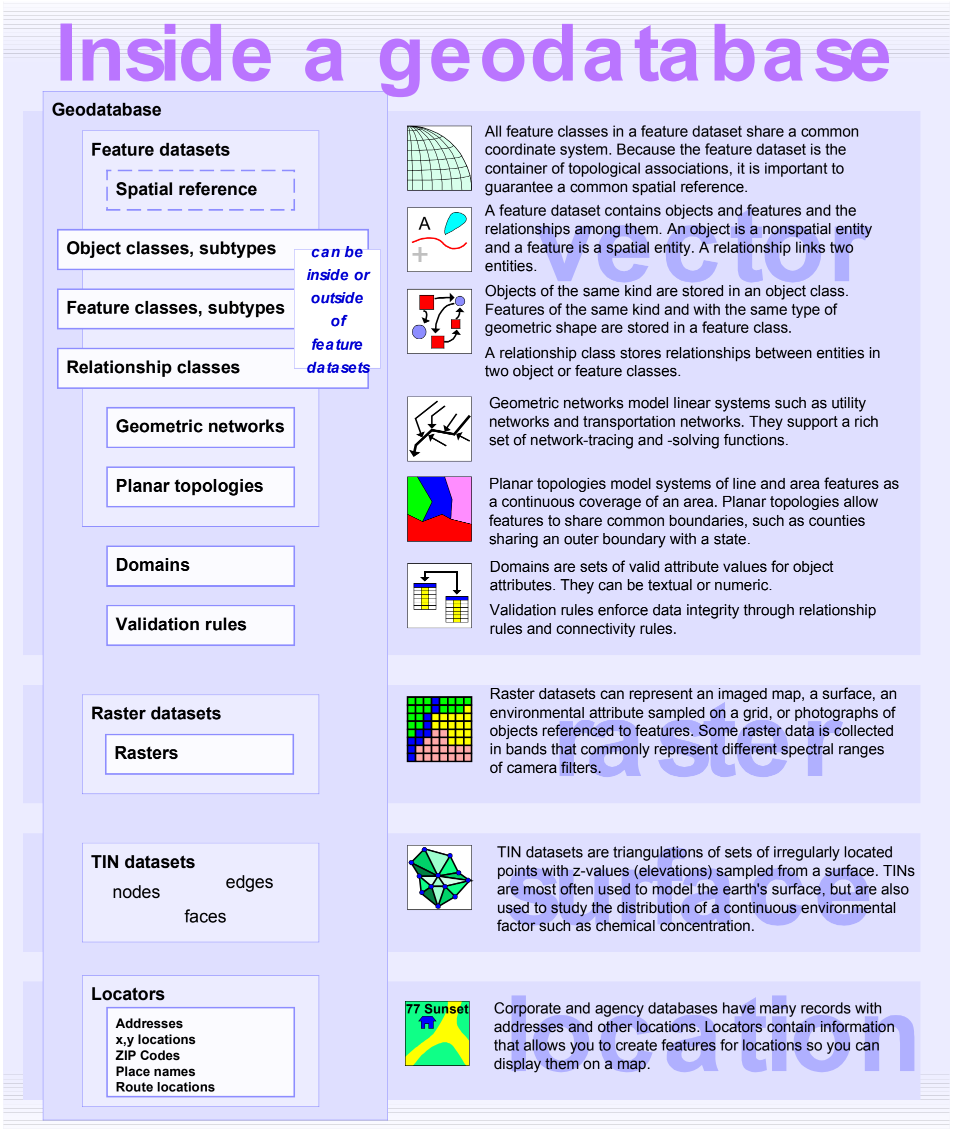


Figure 3 Triangulated Data

A geodatabase implements the vector data representation with feature datasets and feature classes, the raster data representation with raster datasets, and the triangulated data representation with triangulated irregular networks (TINs).



Inside a geodatabase

Geodatabase

Feature datasets

Spatial reference

Object classes, subtypes

Feature classes, subtypes

Relationship classes

can be inside or outside of feature datasets

Geometric networks

Planar topologies

Domains

Validation rules

Raster datasets

Rasters

TIN datasets

nodes

edges

faces

Locators

Addresses
x,y locations
ZIP Codes
Place names
Route locations

All feature classes in a feature dataset share a common coordinate system. Because the feature dataset is the container of topological associations, it is important to guarantee a common spatial reference.

A feature dataset contains objects and features and the relationships among them. An object is a nonspatial entity and a feature is a spatial entity. A relationship links two entities.

Objects of the same kind are stored in an object class. Features of the same kind and with the same type of geometric shape are stored in a feature class.

A relationship class stores relationships between entities in two object or feature classes.

Geometric networks model linear systems such as utility networks and transportation networks. They support a rich set of network-tracing and -solving functions.

Planar topologies model systems of line and area features as a continuous coverage of an area. Planar topologies allow features to share common boundaries, such as counties sharing an outer boundary with a state.

Domains are sets of valid attribute values for object attributes. They can be textual or numeric.

Validation rules enforce data integrity through relationship rules and connectivity rules.

Raster datasets can represent an imaged map, a surface, an environmental attribute sampled on a grid, or photographs of objects referenced to features. Some raster data is collected in bands that commonly represent different spectral ranges of camera filters.

TIN datasets are triangulations of sets of irregularly located points with z-values (elevations) sampled from a surface. TINs are most often used to model the earth's surface, but are also used to study the distribution of a continuous environmental factor such as chemical concentration.

Corporate and agency databases have many records with addresses and other locations. Locators contain information that allows you to create features for locations so you can display them on a map.

Figure 4 Geodatabase basics

IV. The Geodatabase

ArcGIS is fundamentally based on a new data model called the geodatabase (short for geographic database). The geodatabase has two major concepts. First, a geodatabase is a physical store of geographic information inside a DBMS. Second, a geodatabase has a data model that supports transactional views of the database (versioning) and also supports real-world objects with attributes and behavior (intelligent features) [3]. Behavior describes how an object can be edited and displayed. Behavior includes, but is not limited to, relationships, validation rules, subtypes, and default values.

Three key features of a geodatabase are:

a. Centralized management of a wide variety of geographic information in a DBMS
b. Versioning that allows simultaneous editing by multiple editors and transactional views of the geodatabase
c. Custom (or intelligent) features that have properties, behavior, and relationships

Centralized GIS Data Management

The first key feature of the geodatabase is the centralized storage of a wide variety of geographic information in a DBMS. This is the natural solution for handling the growing amount of digital spatial data and also meets the need to leverage this data among the growing numbers of GIS users. The geodatabase supports multiple formats of spatial data including:

__ Simple file features (such as shapefiles)
__ Custom features with business logic and editing rules
__ Attribute data
__ Metadata
__ Images
__ Raster/Grid data
__ CAD data

Versioning

The second key feature of the geodatabase is *versionin*g. Versioning allows simultaneous editing by multiple users and also allows transacted views of the geographic database. This framework lets you create versions of a geodatabase for the states of a project, reconcile differences between versions, and update the master version of a geodatabase.

Intelligent Features

The third key feature of the geodatabase is that you can add intelligence to features in the form of behavior, editing rules, and relationships. In addition to modeling generic features such as points, lines, and areas, you can create features that more closely resemble real-world objects such as landmarks, aids to navigation, and limit areas. Features with relationships and rules are known as custom features. Custom features can be created without any programming using ArcGIS. Optionally, you can use computer-aided software engineering (CASE) tools to create custom features and a geodatabase schema from a Unified Modeling Language (UML) diagram. The essential idea is that an intelligent database contains business rules and business objects. As a result, GIS applications require less customization and programming out of the box.

Geodatabase Topology

The geodatabase maintains topology in feature data sets. Feature data sets are collections of feature classes that share a common spatial reference (coordinate system) in order to achieve topological associations.

Supported Formats

ArcGIS supports a wide variety of data formats [3].

GIS Data Formats:			
Shapefiles	DIME	GIRAS	SDTS
Geodatabase	DGN	IGDS	SLF
ArcInfo Coverages	DLG	IGES	TIGER (thru v2000)
Shapefiles	DWG	MicroStation	VPF
ADS	DXF	MOS	
DFAD	Etak	S-57	

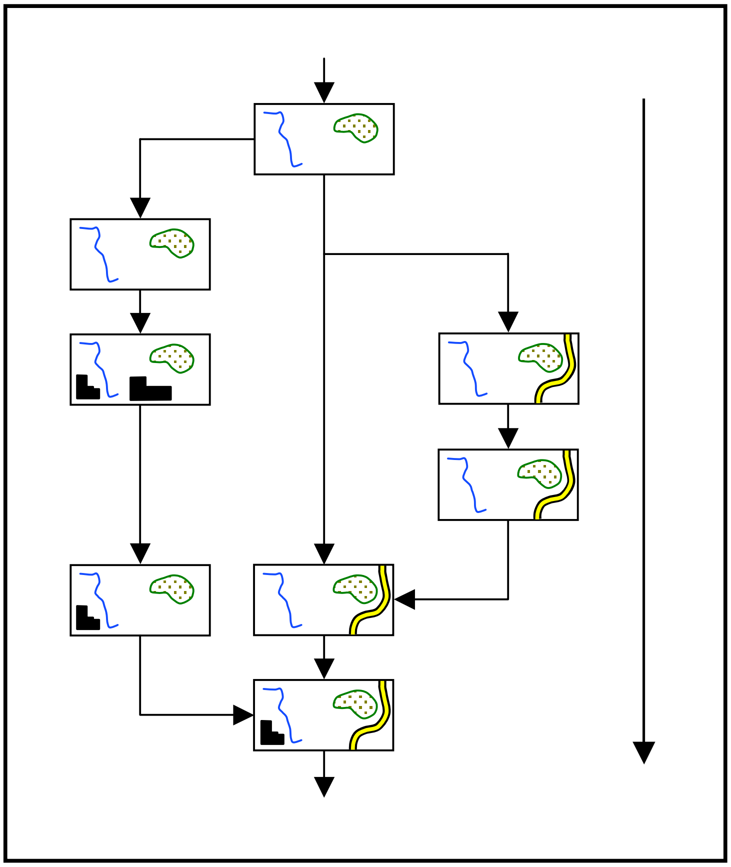
Raster Formats:			
ESRI GRID	DTED Level 1 and 2 (.dt1)	National Image Transfer Format (NITF) (.ntf)	ADRG
ESRI SDE Raster	ER Mapper (.ers)	Portable Network Graphics (.png)	BIL
ESRI Raster catalogs (Image Catalogs)	ADRG Image (.img)	LizardTech MrSID (.sid)	BIP
ERDAS IMAGINE (.img)	ADRG Overview (.ovr)	Tagged Image File Format (TIFF) (.tif)	BSQ
ERDAS 7.5 LAN (.lan)	ADRG Legend (.lgg)	Raster Graphics (CADRG) (.crg)	BMP
ERDAS 7.5 GIS (.gis)	Sun Raster	Windows Bitmap (.bmp)	DTED
ERDAS Raw (.raw)	TIFF	JPEG File Interchange Format (JIFF) (.jpg)	ERDAS
ESRI Band Interleaved by Line (.bil)	ESRI GRID Stack File (.stk)	Controlled Image Base (CIB) (.cib)	GRID
ESRI Band Interleaved by Pixel (.bip)	RLC	ESRI GRID Stack (<directory>)	GRASS
ESRI Band Sequential (.bsq)	IMAGINE	Graphic Interchange Format (GIF) (.gif)	JFIF

Additionally, other raster formats can be supported by modifying the DLL filter file to add file names as keywords that allows the entry to be recognized as a proper entry and adding additional information to the raster format configuration file.

V. Data Storage

Seamless Database

The Geodatabase offers the cartographer the ability to hold data within a seamless database. This is often referred to as a Master Feature Database (MFD) [2]. Holding data within a seamless database holds benefits to both the cartographer and the end user. A seamless database if built correctly, can support both multi-scaled data, and multiple products [2]. Units of work can be divided up into more logical work packages, not constrained by artificial chart boundaries. Operations, like edge matching, that is required when compiling charts using a sheet based system are eliminated. Data can be viewed in its true geographic state, i.e. a limit area that crosses a sheet edge is no longer split in two but can exist within the database as one feature. This reduces storage of duplicate attributes, eliminates the possibility of a mismatch in attributes and enhances the information held within the database.



Version Management

Multiuser GIS requires DBMS and strong GIS tools to work with the geographic data. As your organization evolves, your spatial database will grow in size and in number of users. Using a DBMS is a practical means for sharing and protecting your data investments.

Admittedly, the use of DBMS in GIS today is not widespread. For years file formats have been used to manage and share information using coverages, shapefiles, grids, TINs, CAD drawings, and many raster formats. There were some clear advantages to doing this. First, it was easy and inexpensive. Everyone could edit and manage their data this way. No DBMS investment was necessary. Plus, DBMSs lacked the ability to manage the larger, more complex data types and support for operations required by GIS (for example, to execute a single query to fetch a few thousand features for display in a particular map extent or to edit features using a long transaction that can last for many days and even weeks) [2]. As DBMS has evolved into more powerful technology that could support complex data objects, larger queries, and stronger transactional support, the use of DBMS for geographic data management has become much more practical [4]. Most multiuser GISs today manage their tabular data in a DBMS, and some have begun the migration of their geographic information into a DBMS. This migration from file-based data to DBMS will continue in the next few years. Like other information resources in an organization, GIS data can benefit from the addition of a DBMS. It makes good business sense to manage your GIS information resource as part of your relational database. Some of the reasons to use relational databases are as valid for GIS as they are for other information:

- Concurrency management in a multiuser environment
- Standard data management practices such as backup, recovery, and replication
- Performance for any number of users
- The need for managed and organized data
- Scalable data volumes with no size limitations
- Centralized system-wide or company-wide access to the data
- Data maintenance over long time periods, spanning personnel changes and hardware/ software upgrades
- System failure/recovery mechanisms
- True client/server and Internet architectures

Metadata

Metadata is information that describes your data in the same way a card in a library's card catalog describes a book. When a data source has been properly documented, metadata can answer your questions and help you make decisions. For example, it can help you determine when your data is out of date, what map scale is appropriate for presenting the data, or how accurate your data is and therefore how much confidence you can have in your conclusions.

Metadata in ArcGIS consists of properties and documentation. *Propertie*s, such as the extent of a feature, are derived from the data source itself. *Documentation* is descriptive information supplied by a person [5]. Metadata information is therefore multileveled from the entire geodatabase level down to the individual feature.

You can enter descriptive information about the data source by the metadata editor. You might provide details about feature attributes such as name definitions, how the data was collected, and valid values [5]. Documenting your data protects your organization's investment in that data.

Some documentation may already exist in a text file or as a Descriptive Report document. As a preliminary measure, you can attach a copy of the file to the data source's metadata just as you would attach a copy to an E-mail message.

Documentation standards

The International Standards Organization (ISO) is attempting to create a unified content standard. With a common format and set of terminology, it will be easier to document data and share information with others. ArcGIS metadata is compliant with the FGDC standard *(Content Standard for Digital Geospatial Metadata)*. Using the FGDC editor will create FGDC-compliant metadata if the prompted documentation items are completed.

How metadata is stored

Metadata created with ArcGIS is stored as XML data either in a file alongside the data source or within a geodatabase. Metadata files become a part of the data source itself. They are automatically moved, copied, and deleted along with the data source.

Customizing metadata

Programmers can create custom metadata editors and add them to the ArcGIS application. A custom editor can generate metadata according to a different standard or define custom content for your organization. For example, a company could add elements recording business procedures such as whether or not a data source has passed a quality control review or maintain individual feature histories.

VI. Cartographic Considerations

Topology and Database Validation Rules and Methods

Topology is used to model the spatial relationships between features.   These topological characteristics between object classes are set out within the schema.   Although not required for the production of paper maps, digital data normally requires topology for analysis and calculations.

Until recently, digital map data has been captured using points and lines with areas simply inferred in the symbology [2]. Demand for data using areas has increased, with the acceptance of standards like the Vector Product Format (VPF) and the Electronic Nautical Chart (ENC) S57 profile. By using topology, it is possible to create automatically these areas from the line work, and as a result of the shared geometry the areas are maintained as the lines are edited.

The database designer can also specify integrity constraints on the database by setting up validation rules and methods.   This is a powerful way of ensuring the data within the database meet pre-determined business rules and constraints. The user can set validation on objects either with validation rules, or by writing object methods. The validation rules are easy to set, via a graphical user interface, and support the consistency between the attributes of one object, attributes between multiple objects and between attributes and the spatial placement of another object.

A validation rule might be as uncomplicated as the value of attribute Color can only be blue or black. Alternatively, they could be more complex such as the class Contour cannot cross a feature of the class Lake and cannot intersect with itself or another feature of class Contour.

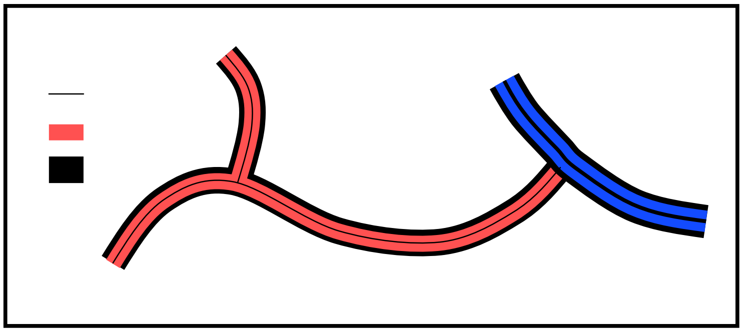
Representation and Symbolization

There are a number of ways that data can be represented on screen or in a hardcopy output format, ranging from the simple setting of a line style/color, through to a complex drawing method that exhibits dynamic representation [2]. Displaying all features of a single class using the same symbol is the simplest form of data representation. Users can create their own style sets or customize the standard ones that are provided out of the box.

The representation of a feature class can be controlled by a unique attribute that exists on that feature class. If for example, the feature class was road, and it had an attribute called type, this attribute could be used to control how the feature is represented. This means that as the attributes of features change the representation automatically reflects this change.

The third type of supported representation is a graduated color ramp for use on maps that are primarily comprised of numeric data with a progression or range of values like elevation. By generating statistical information from the data, this can be used to assign values from the color ramp. The user selects the attribute that will be used to generate the progression or range then chooses the type of classification (Manual, Equal Interval, Defined Interval, Quantile, Natural Breaks, Standard Deviation). If required, a second attribute can be used to normalize the data.

These last two methods of feature representation use a lookup mechanism on certain attributes, which means that features display always shown their latest state. This type of changing representation is often called dynamic representation [2].



To support complex cartographic representations all feature classes can have multiple parts associated with each symbol and the draw order of the feature classes can be controlled [2]. Adjacent is an example of this multi-part representation, all three components for both feature classes are produced from the one piece of geometry, but by drawing the parts in the correct order a road casing representation is produced. The diagram also illustrates that part priority is honored between feature classes. This is also the case between features in the same class.

The most flexible form of representation for features is a display method.   This method is called on an object when it must draw itself. How the object draws (color, symbol, shape, size, etc.) is all controlled from within the method. By customizing the behavior, the display of one object could take into account its spatial relationships with other objects, or it could use a combination of attributes to determine how to draw itself.

The diagram below shows an area feature, Mud that is using two types of fill patterns. The type of fill depends whether the Mud feature is over a Sea or Land feature. The display method is controlling the drawing of the Mud feature, hence if either the Sea or Land features are edited the Mud feature will pick up these changes and draw itself accordingly [2]. The boundary between the land and sea areas that is used by the Mud display method does not exist as a geometry within the database, but is created from the land and sea feature geometries. This form of active representation greatly reduces the costs of data maintenance, since an edit to one feature is reflected in the other objects automatically.

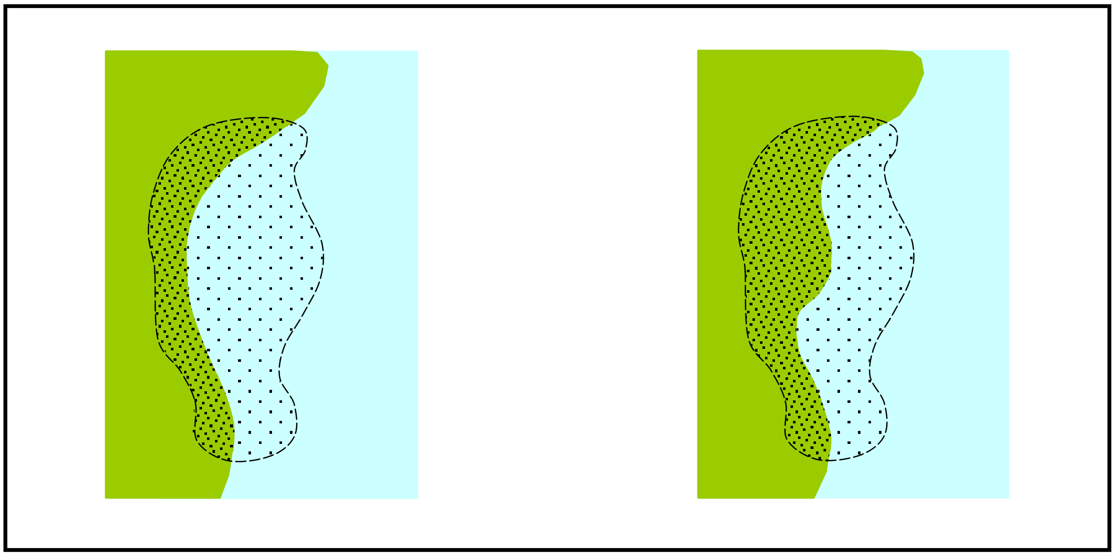
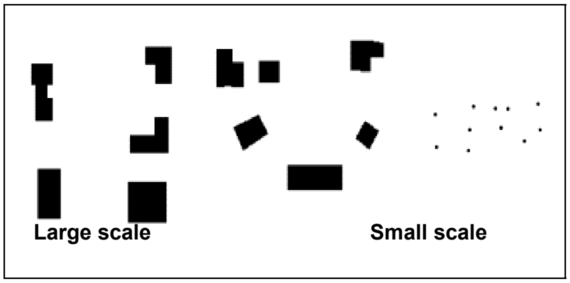


Figure 5 Dynamic representation with topology and a drawing method

Display methods can also be used to perform on-the-fly generalization of a feature's geometry dependent on the scale of the map. In its simplest form, this might mean that an area feature, such as a city, may be represented as a collection of individual buildings and street networks at large scales, a fill pattern or tint area at smaller scales, to a point symbol as the scale of the map is decreased. Generalization that is more complex might entail simplifying the boundary of buildings as the scale decreases, or possibly aggregating objects using feature associations to control the objects that can be aggregated [1, 2].



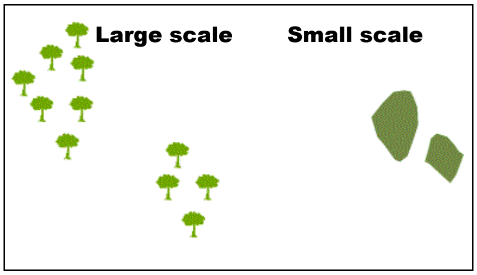


Figure 6 Building Representation

Figure 7 Aggregating Objects

Feature Labeling

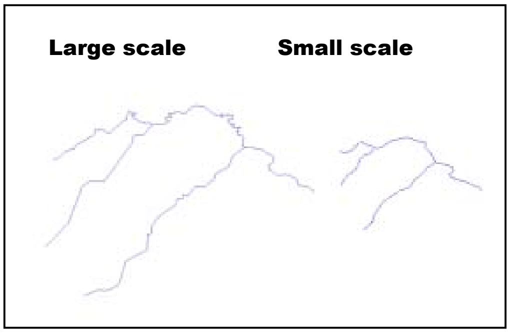
An advanced feature labeling engine is included with the ArcGIS applications. This facility removes much of the user intensive editing associated with the placement of feature annotation. The labeling engine allows the user to set the placement of a label with respect to the source object and any clashing objects. The user can also select whether to link the label to its associated object, the font to use and even define an expression that will be used to generate the label text. By defining an expression, the user can combine several attribute values or perform some form of conditional statement when forming the label string [2].

Generalization

Data are entered into the Geodatabase, with reference to their source scale, 1:10 000, 1:50 000, etc. It is possible, using the extensive representation and symbolization tools available, to produce multiple products from the same seamless (master feature) database. In order to support multiple products at different scales from the data's source scale some generalization of the data may be required.

Generalization tools that perform solely on a feature's geometry (remove extra vertices) are of little use to the cartographer because the context of the feature is often lost. A good generalization algorithm considers this context when simplifying the line work and generates a new geometry that is appropriate for the feature [2].

Figure 8 Stream feature generalization



VII. Data Dissemination

A geodatabase accessible to an enterprise internet or the world wide web can receive thousands or even millions of requests each day, and each request might apply the data to a completely different application. To manage such a load, "middleware" is needed between the geodatabase server and the web server connecting to it to the Internet. ArcIMS (Internet Mapping Server) provides this capability for ESRI.

The ArcIMS architecture

ArcIMS runs in a distributed environment and consists of both client and server components. The ArcIMS server components allow you to extend an Internet Web site by including GIS capabilities. Your ArcIMS Web site can deliver GIS data, maps, and applications.

With ArcIMS, you design and author a GIS Web site. Many types of client applications can connect to and use your ArcIMS Web site over the Internet, ranging from the free HTML and Java viewers included with ArcIMS to more advanced ArcGIS Desktop applications.

The ArcIMS architecture is highly scalable. Many concurrent users can access your Web site, and many requests can be processed at one time. With ArcIMS, you can expand your server over time to accommodate growth in demand. Because of its scalable architecture, ArcIMS is as good for small organizations serving dozens of maps a day as it is for large organizations serving hundreds of thousands of maps a day [4].

ArcIMS services

When an ArcIMS client sends a request to a server, the ArcIMS server processes and responds to the request. Typical ArcIMS requests generate maps or retrieve geographic data for a given map extent. The most common ArcIMS service is a MapService.

MapServices

When you create a map for your Web site, you determine the data layers to be included and the way in which map features will be rendered. You define symbology, add labels, set scale factors, and so on. When a client sends a request for a map, it is generated on the server based on your specifications. A MapService delivers the map to the client in one of two ways: as an image or as streamed features. An Image MapService uses the ArcIMS image rendering capabilities to deliver a snapshot of your map to the requesting client [4]. The snapshot is sent as a compressed JPEG, PNG, or GIF file. A new map image is generated each time the client requests new information. The Image MapService can also deliver compressed raster data to clients. A Feature MapService streams compressed vector features to the requesting client [4]. Feature streaming enables more advanced clientside tasks such as feature labeling, feature symbolization, map tip creation, and spatial selection of features. This functionality allows the user on the client side to change the appearance of the map.

Depending on your organization's objectives and the audience for whom you are building your application, you may want to incorporate additional GIS capabilities into your Web site. Additional GIS capabilities available with ArcIMS include data query and data extraction.

Data Extraction and Querying

You can build new queries or run predefined queries to derive specific information about the data. The client submits the query to the server, and the server returns query results to the client. You can also extract geographic data from the server. The server responds to a request for data by sending zipped files such as shapefiles to the client. The user who receives the data is then able to extract the zipped shapefiles to a local drive.

VIII. Standards

ArcGIS has been designed and built using software component architectures including Microsoft Component Object Model (COM) technology for desktop and local network applications and Java for web based applications. These standards have proven themselves in the marketplace as a result of their openness, functionality, scalability, and adaptability to integrate with other IT software.

The concept of components is to divide software functionality into modules or independent pieces that can be developed, tested, and combined into complete programs and ultimately reused [1]. The benefit of software component architecture is improved software quality, better performance, and the ability to more quickly update software versions and functionality.

Finally, the benefit of implementing open standards and interoperability is to leverage technology from other industry segments, such as object-relational database management systems, and to allow GIS companies to concentrate on research and development on core GIS functions.

IX. Conclusions

GIS technology advances over the past several years provide effective and efficient data management for processed hydrographic data. The geodatabase model lends itself well to database designs and applications associated with the collection, documentation, distribution, and analysis of large amounts of vector, raster, and surface modeling data (i.e., a data warehouse). It also provides the end user with the options of setting states and behaviors for each data object. Using this approach provides the means of storing, analyzing and modeling ocean data and enabling the existence of "intelligent data objects" and methods to distribute value added hard and soft copy products to end-users.

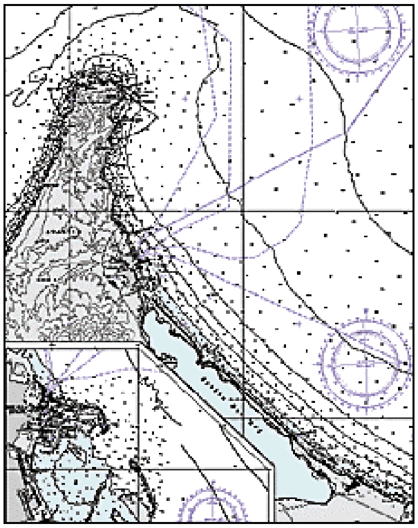


Figure 9 DNC Hardcopy chart by ESRI

X. References

[1] Zeiler, Michael, *Modeling Our World, The ESRI Guide to Geodatabase Design,* ESRI Press, Redlands, CA, 1999.

[2] *ArcInfo 8 Cartography, New Possibilities,* ESRI White Paper, ESRI, 1999.

[3] *ArcGIS-One World. One GIS*, ESRI White Paper, ESRI, 2001.

[4] *What is ArcGIS*, ESRI White Paper, ESRI, 2001.

[5] *Using ArcCatalog*, ESRI White Paper, ESRI, 1999.