

# MODELING ACTIVE DATABASE-DRIVEN CARTOGRAPHY WITHIN GIS DATABASES

Charlie Frye and Cory L. Eicher

ESRI, 380 New York Street, Redlands, CA 92373

## Abstract

GIS databases contain classes of features that are representations of real world phenomena; and GIS and mapping software provide a means for encapsulating that information within a map. Typically, ancillary cartographic information like legends, scale bars, graticules, and text have been stored outside of the database within documents or as instructions in a program that reconstructs the map each time it is needed. As more mapping agencies and organizations move in the direction of centralized data holdings, the decentralized nature of traditional cartography will cause inconvenient management constraints. ESRI has spent the last year developing technology for storing cartographic information within a traditional centralized DBMS structure. This allows cartographic information to be managed far more effectively than current methods permit. One of the chief values of this technology is the ability to keep cartographic products constantly up to date, i.e., each time a map is produced the most current information is used. This technology also allows for changes to the cartographic standard to be instantly applied to all products that use the standard. In essence, we are exploring making the leap from database-derived cartographic information (passive) to database-driven cartographic information (active).

## 1.0 Introduction

This paper outlines an approach for using the data management structures and conventions that are available in most database management systems (DBMS) to manage and use cartographic information. Two definitions of cartographic information are presented, and then, based on the broader of these definitions, a conceptual data model is presented for storing cartographic information in a relational database. This conceptual model is the primary focus of this paper. To support this presentation of this model, definitions are provided for various necessary data structures and management conventions. Furthermore, these definitions presume the larger context of a geographic information system (GIS). In this model, a GIS database is the container for both GIS data and cartographic information. Specifically, ESRI's geodatabase(1) is used. This paper does not seek to define a physical data model for storing cartographic information in a GIS database, though this is certainly a direction for future work.

References to, "conceptual data model" (2) and "physical data model" (3) will be used throughout this paper. A conceptual data model, or conceptual model, is logical and general; it is intended as a first stage in a data modeling process. A physical data model relates exactly how specific kinds of information are modeled, stored, and managed within a database. A physical data model is one of the results of a data modeling process.

At ESRI, the data modeling process has followed a fairly traditional path of starting with a goal, understanding the logical components and their interrelationships that will be used to achieve the goal, articulating the physical storage and database management model that will achieve the goal, and finally proving the model with robust case studies. This process is used for deriving data models for the proprietary functionality of the geodatabase and for recommendations for how customers model their GIS data in order to best take advantage of the geodatabase. This same process is the approach taken in the current paper.

### 1.1 Cartographic Information

Cartographic information can be characterized as having two forms. The first form is geographic data prepared for cartographic use, usually through a batch or manual editing process applied to GIS data. Cartographic information in this form could be characterized as cartographic features that are produced for a specific purpose or map. The second characterization of cartographic information is more generic and includes everything else that goes on a map, including: the page definition, rules for map composition, definition of the colors and symbols used to represent the data on the map, and the processes and sequencing necessary for making the map.

The underlying data model for the first form of cartographic information—cartographic features – is already well proven, utilizing the same management system as for storing GIS features. To be clear, this means storing geographic

features in spatially continuous classes of that are thematically homogeneous. This assumption about the nature of geographic feature storage is critical because managing older CAD, tile-, or sheet-based data requires the unnecessary repetitive application of data management tools that often result in poor organizational and data update performance. Such approaches also prohibit using the data for mapping applications that allow any geographic location to be used as the center of the map.

Using the geodatabase also provides a dramatic organizational advantage, that is, the ability to use a versioned database that allows multiple users to be simultaneously editing the data. Beyond the obvious workflow advantages, such as being able to focus resources on critical geographic areas, the geodatabase also affords organizations the ability to leverage in-house DBMS management and administration skill sets that are often already well established within an organizations information technology (IT) infrastructure. This is because the geodatabase operates using data stored in standard RDBMS technology like Oracle®, Informix®, IBM® DB2®, and Microsoft® SQL Server™.

Figure 1. Common Configuration for Multi-user Versioned Geodatabase

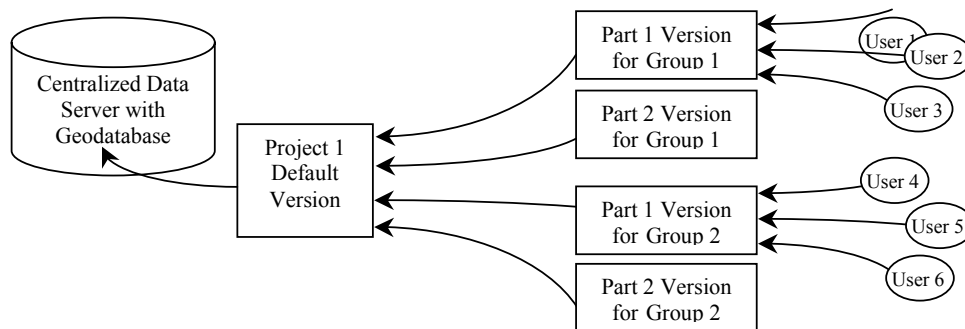


Figure 1 shows an example of how versions can be organized within a geodatabase to facilitate work by groups of users. The arrows represent the flow of changes to data as they are:

1. Made by the users
2. Reconciled in the versions for each group and part
3. Reconciled into the project's default version used by a production team to make maps.

So, while the importance of understanding how to model cartographic feature information is indisputable, this paper focuses on modeling the lesser understood second form of cartographic information described above. Within this scope, the following table describes the major types of cartographic information that must be managed within the database.

Table 1. Major Types of Cartographic Information

Major Type	Description
Colors	The list of all colors used on the map.
Page Definition	Describes size of the page, rules to locate parts of the map and how those parts should appear. Also included here is the color model for publishing the map.
Symbols	Specifies how a given geographic feature or map element is to be drawn on the map. For example, a line symbol would minimally have a color from the list defined above, a width, and a cap and join style.
Symbolization Rules	Describes how cartographic or GIS features are to be symbolized. Includes symbol assignment as well as a description of how information is layered on the map, including masking, trapping, and other processes that contribute to the map's graphical information model. This also includes text for labeling cartographic features.
Series or Location Index	Describes exceptions to the page definition or symbolization rules that may occur on given map sheets or for given map locations.
Map Process Model	The set of processing instructions needed to transform the raw data for the map into a finished product. This includes any geoprocessing functions that are applied to GIS data and encompasses the entire set of steps and set the sequence of those steps. Geoprocessing functions manipulate spatial data and range from traditional GIS oriented functions like intersect, union, identity, etc. to very specialized functions like those used for cartographic generalization.

One item of note in Table 1 is that because the Series or Location Index may override symbolization and page definition properties, overrideable properties and processes for updating must be accounted for within the physical data model and expressly managed in the database.

### 1.2 Database Management Techniques

A myriad of data and database management techniques are widely available and could be used for the management of cartographic information. In a database, objects that are derived from the conceptual model are stored as rows in tables, and objects that share similar kinds of properties are stored in the same table. Just as geographic objects (e.g. features) in a GIS database have spatial and non-spatial relationships to one another, so do cartographic objects. Relationships and indexes can be used in a database to manage how cartographic information is used to create maps, just as these same database constructs are used to maintain operability in a traditional GIS.

### 1.3 The Geodatabase as Context

The geodatabase is also used as the context for this work because it affords a number of specific advantages. Because the geodatabase is a layer of homogenizing functions that overlay a number of specific commercial DBMSs it can be said that the model put forth in this paper will also work on those DBMSs. The geodatabase also has or has under development a number of higher-order object types that directly facilitate modeling cartographic information. The most useful of these objects types are geoprocessing functions, geoprocessing tools, and geoprocessing models which are part of ESRI's ArcGIS 9.0 release. These objects allow data processing and map building functions to be encapsulated and sequenced within a tool or model and applied as they are needed. Geoprocessing tools and models are ways to encapsulate geoprocessing functions and use them effectively within data processing workflows. Specifically, geoprocessing tools are used in ways analogous to how geoprocessing functions are used; the tool form simply has a user interface to set up inputs and parameters. Geoprocessing tools are distinctive from functions in one additional way in that they can be included in a geoprocessing model. Geoprocessing models are collections of logically organized tools that perform a specific sequence of geoprocessing functions. In data processing workflows, geoprocessing models serve roles that are very similar to AML (ArcInfo Macro Language) scripts for processing ArcInfo® data.

### 1.4 A Map as Context

We assert that to develop a proper solution to the challenge of storing cartographic information in a database, one must consider a map not as the finished product, but instead as the process and means for producing the finished product. In fact a map in the context of this paper is actually the *framework* for producing a map, given requirements for producing a specific map product. This framework encapsulates the cartographic specification and the process for creating the map, while the design, the look, and the use of the map are independent of the framework. This framework, by design, supports the creation and production of a wide variety of cartographic designs, even those with different end uses, while using some of the same GIS data as inputs.

In a sense, this paper describes the beginnings of a framework for multi-representational, and/or multi-scale cartographic databases. Though this paper focuses on the data model for such a database, a smart map (4) is another ultimate goal. A smart map essentially is a model for producing a map that knows how to stay up to date given updates in the underlying or input data. Future research will be directed toward the procedures for automatically updating data, finding areas of maps that need to be updated, and finally automating the procedures that will update the maps.

## 2.0 Toward the Physical Data Model

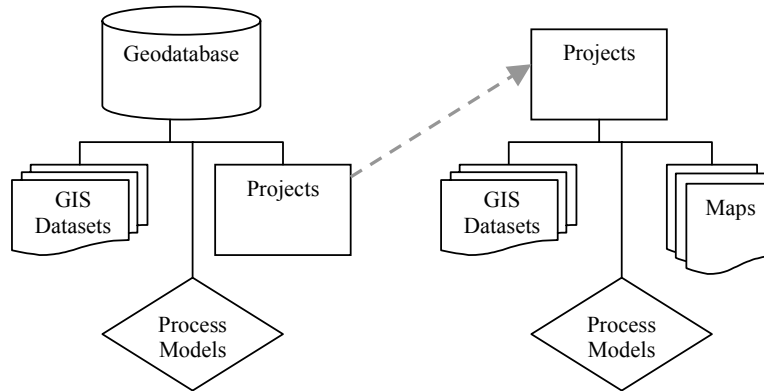
While this paper does not propose a specific physical data model, it does describe a detailed conceptual data model in an articulate manner that is consistent with how a physical data model for this data might look. Items missing include data storage optimizations, i.e., normalization, and to a large extent the role of the application in reading and creating maps from the data.

### 2.1 High-Level Organization of Cartographic Information

Given the concept of a map as described in 1.4 above, the role of this data model is to first capture the essential objects and provide a structure that supports the necessary relationships between those objects. Figure 2 shows that there are four, top-level essential object types in a map: GIS Datasets, Maps, Tools & Process Models and Projects.

GIS datasets can be stored at the root level of a geodatabase or within projects. However, maps are only stored within projects. Projects organize the necessary items for a map that will be published regularly. In other words, projects manage everything needed to produce an instance of a given map product.

Figure 2. Top-level object cartographic organization in the geodatabase



### 2.2 Organization of Major Cartographic Object Types

At a finer resolution, there are also several major cartographic object types. These types play a key role in the overall framework because: a) they can be used in multiple projects, i.e. they can be shared and b) they can organize other finer resolution objects.

Figure 3. Organization of major cartographic objects in the geodatabase

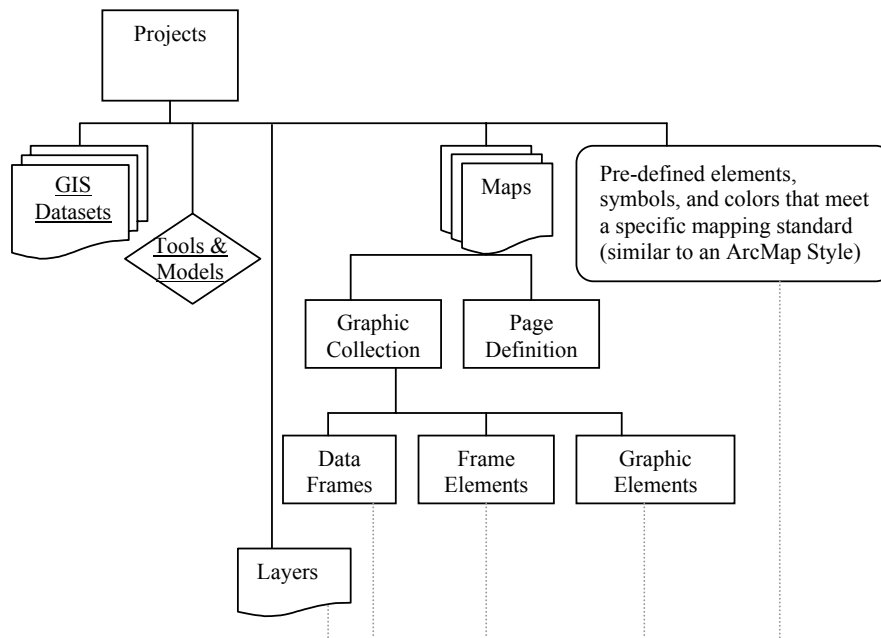


Figure 3 depicts the major cartographic objects and how they relate to one another. Three kinds of relationships are shown. The first is depicted with a solid black line denoting that the object above contains the object below, e.g., maps contain data frames or projects contain maps. The second is shown with underlined text to indicate that this kind of object can exist outside of a project. The third type of relationship is shown with the dashed gray line, and it means that pre-defined colors, symbols, and elements are referenced by any of the connected objects.

Table 2. Roles of major cartographic object types

Object	Description
Map	Includes collections of data frames and elements that are arrayed on a page.
Page Definition	Page size and rules that govern the placement of data frames and map elements.
Graphic Collection	A collection of map elements that are located relative to the page or relative to one another.
Data Frame	The area of the map that shows geographic data. Geographic data is displayed as thematically organized layers of geographic data. Data frames can also display reference systems such as graticules, and they can also broadcast information consumed by frame elements. For example, map scale information is broadcast to scale bar elements, and map content information is broadcast to legend elements.
Frame Element	Map marginalia linked to a data frame that derives key display properties from the data frame. Scale bars and legends are two examples.
Graphic Element	Map marginalia that is not linked to a data frame; including text and non-geographically located graphic entities such as points, lines, and areas.
Layer	A set of instructions for how to display a thematically consistent set of geographic data.
Pre-defined Elements, Symbols and Colors	Together with the page definition and element locations, these items define a map's cartographic specification. Colors are used by symbols, which in turn are used by predefined elements, layers, and data frames.

### 2.3 Map Series and Atlases

A map can show a portion of a larger geographic extent. To facilitate creating map series and atlases, an index layer object type is introduced. An index layer is a geographic data set whose geometry serves to locate the centers or extents of individual maps in a series or atlas. Typically only one index layer is used, though many are supported. Each record in the attribute table of the index layer contains information that a mapping application can use to update the properties of the elements in the map. For example, the index layer may have a string field that contains the titles for each sheet in a map series. It may also contain a scale that should be used for each sheet in the series, and it can indicate the extent of the map sheet on a locator map.

The index layer operates within a variety of map production workflows, including managing data production, assuring quality, and managing output and update processes. For example, at the onset of a new round of publishing, the index layer supports querying to learn which map sheets require updating after more recent geographic data has been added to the database, thus indicating updated areas in some but not all of the map sheets. Also, to support map series and atlas production and management, page elements and data frames are configured to be updated based on the active row in the index layer table. The main enhancement is for text which was previously a static graphic element but can now also be a frame element whose contents are updated dynamically based on the active row in the index layer table. An automatically-changing map title on each page of an atlas is the archetypal application of this construct.

To create a map that can be updated dynamically based on which record in the map index is being used to set the geographic extent of the map, some properties of elements on the page must be specially defined. The main change is to allow text to be treated as a graphic element (static) or as a frame element (dynamic, updated by the contents of the index layer).

### 2.4 Tools and Models

To make data suitable for maps, tools and models are applied to geographic data to produce cartographic representations of that data. A tool is a specific algorithm that is applied to a specified set of data to produce another set of data. Models are a set of tools that may be applied to the layers in a map. Models can also contain other smaller models, so for instance, a model that processes the data for a map may contain a model that smoothes the geometry of some roads and a tool that displaces the roads such that their symbols do not touch or overlap railroads. Models are not limited to processing single data sets. In fact, for cartography, models must support multiple and/or different input and output data sets. For instance, streets may be used as an input parameter for a tool that displaces buildings so that displaced buildings are not placed on top of or too close to the streets. Finally, tools and models can be created to do both very general processing, such as creating buffers around features, where any features could be used as inputs, as well as tasks that are specific to a given dataset. In the latter case, for example, an organization's data may have some special attributes that allow tools to produce a specific, perhaps branded, effect.

## 2.5 Using Feature Attributes to Drive Symbology and Text

The most common way of symbolizing GIS data on a map (including labeling with text) is to derive symbology from attribute values. The same method is recommended here. However, additional guidance is offered for using attributes in the most effective way. Table 3 describes three models for encoding feature symbology – through feature-type definition – in a relational database. These are based on domains which are used to assign additional rules to attribute fields, like the range of values permitted for measured data or the valid values for coded data.

Table 3. Techniques for modeling feature types in attribute data (5) to automatically drive symbology

Model	Description	Usage and Benefits	Information available for cartographic representation
Simple Domain	A single integer field is used to define the legal values that may be stored in an attribute field. These may be used to define the minimum and maximum values for measured data, or they may be used to define a list of legal codes (as integers) and their descriptions.	This model is useful for displaying different types of features with different symbols. Using an integer is faster than using a string – less data is read, transferred, and interpreted. The model is most useful when there are fewer than 100 kinds of features in each feature class.	Each feature is displayed based on its primary or predominant usage type.
Subtype and Domain	Subtypes are groups of objects (rows) in a GIS feature class that are grouped together by an attribute. These can be combined with domains to create a hierarchical classification. Two integer fields are used, a primary type and a qualifier. The first is defined in the database as a subtype. The second qualifies the first by specifying the valid values (i.e., a domain) for each of the subtypes. An independent domain can be applied to each different subtype.	This model is useful when there are between 50 and 200 different kinds types of features in a feature class and when there is frequent redundancy in the naming of feature types.	Allows for two-level descriptions, like Stream – Intermittent, or Lake – Fresh Water.
Complex Domain	Like the Simple Domain model, this model uses a single integer field in the feature class, but this integer value is used as a key in a relationship to several other related tables that provide a hierarchy of descriptions. This model was first developed for NIMA (6), and it is the general style of that model that is advocated here.	This model is useful when there are potentially thousands of kinds types of features. Creating such a model forces semantic integrity, which cannot necessarily be said for the other two models.	Allows for multi-level descriptions that can include multiple uses for a given feature. For example, “building” may be a feature’s primary type, but it may function as a Methodist church, community center, and synagogue. Such a feature would be described in this model with “Building, Methodist Church, Synagogue, Community Center” and then symbolized and/or labeling by the mapping application according to this information.

## 2.6 Cartographic Data for Multiple representations (scales)

Multiple representations of the same GIS features for different map scales or for different map products is an issue that concerns many mapmakers today. Jones (7) shows that extensive research on the form and schema of a multi-scale system is indeed a daunting problem. However, at present the problem of producing and managing such data is mostly one of effort. A GIS database can include many representations of the same features, and a mapping application can be set to show specific data at specific scales. The challenge lies in making the effort to establish and manage the relationships between the derived cartographic representations of features and their source GIS features. The management task is especially hard because each tool that may be applied to some GIS features has specific input requirements. These requirements must be captured such that the resulting cartographic representation of the data can track or be informed of relevant changes to the source data. It is safe to say that the full set of cartographic data production tools does not yet exist in a digital environment yet; therefore, the full set of potentially relevant parameters to track does not yet exist. However, many relevant parameters are well understood. For instance, if the geometry of a

GIS feature is changed, it should notify any dependent cartographic features that were produced based on this geometry. This notification at a minimum can set a “dirty” state for the dependent cartographic features.

Automatically and intelligently cleaning up all ‘dirty’ cartographic features is something of a holy grail, but there are conceivably some scenarios where a tool can be re-run on dirty features to see if a new acceptable cartographic feature can be produced automatically. This formally introduces one final concept, *cartographic features*. These are features that have knowledge of their symbology and participate in map-specific topological relationships with the other features on the map, including text. Thus, a combination of feature type-specific update-warnings and map-specific topologic integrity rules can produce and validate then reproduce and re-validate cartographic data.

### 3.0 Conclusions

Today, most cartographic information can be modeled in a GIS database. The primary exceptions are data with complex relationships that are not yet well understood, like those contained within multi scale or multi representational systems. Although at present it may not be particularly easy or intuitive to model cartographic information in a database, this is beginning to change as many mapping organizations continue to find new benefits for storing their spatial data in a GIS database. These same organizations will benefit further by using these systems for producing and managing maps. These organizations in turn have begun to exert pressure on GIS software companies to produce cost effective solutions for map production from geographic databases.

The proposed conceptual model put forth in this paper is the result of several years of research on both common GIS data modeling techniques and similar studies of the anatomy of map from a data management perspective. It is expected that this proposal will evolve, perhaps substantially, as more research is conducted and as prototype databases are compiled and tested.

### Bibliography

<sup>1</sup> M. Zeiler, *Modeling Our World*. ESRI Press. (1999)

<sup>2</sup> DataModel.Org, <http://www.datamodel.org/DataModelDictionary.html#CDM> (2003)

<sup>3</sup> DataModel.Org, <http://www.datamodel.org/DataModelDictionary.html#PDM> (2003)

<sup>4</sup> C. E. Frye, Intelligent Multi-Scale Cartographic Data and Their Databases. Cambridge Conference paper No. 2 (2003)

<sup>5</sup> C. E. Frye and C. L. Eicher, Building Cartography into GIS Data Models. Association of American Geographers Conference. paper session No. 6640 (2003)

<sup>6</sup> Digital Geographic Information Working Group (DGIWG) DIGEST Overview, <http://www.digest.org/Overview2.htm> (1997)

<sup>7</sup> C. B. Jones, S. Zhou, M. Lonergan, A. I. Abdelmoty, D. B. Kidner, and J. M. Ware, Multiscale Spatial Databases, <http://www.cs.cf.ac.uk/user/C.B.Jones/MultiScale.html> (2003)