INTEGRATION OF A GIS AND A SEMANTIC DATABASE SYSTEM [*†]

Tin Ho, Christian Pesantes, Venkat Maddineni, Elma Alvarez,
Nagarajan Prabhakaran, David Barton, Napthali Rishe, Jennifer Fu, Scott Graham.
High-Performance Database Research Center
School of Computer Science
Florida International University
University Park, Miami, Fl 33199
(305) 348-1706 Fax: (305)348-1705
hpdrc@cs.fiu.edu ♦ http://hpdrc.cs.fiu.edu

ABSTRACT

Database Management Systems (DBMS) are implemented to allow for storage, and quick retrieval of large amounts of information. With the great demand for Geographic Information Systems (GIS), an effective database is in great need. Choosing an effective database for the GIS is very important because the overall efficiency of the GIS is affected by the database. A GIS consist of two types of datasets: the spatial data, and the attribute data. The attribute data consists of textual strings, which can be related to the spatial dataset. It is here, for the two datasets where the databases can be used. In commercial GIS the spatial data are mostly stored in an internal database. The attribute data can be stored either internally or through an external database. This paper will demonstrate how the attribute data will be stored in the Semantic Database (SDB). Furthermore, a schema has been designed that can store spatial data in the SDB. Possible methods on integrating this SDB schema with a GIS will be explored. The successful integration of a GIS with the SDB would benefit users who currently incorporate GIS with relational databases.

1.0 INTRODUCTION

There has been a remarkable surge of environmental, agricultural, scientific, and academic interest in Geographic Information Systems (GIS) since their graphical nature allows planners to easily visualize the data, which aids in decision making. A GIS is a sophisticated computer based mapping and information retrieval system, consisting of three primary components: a powerful computer graphics program, a set of analysis tools, and one or more databases which serve as the data repository. All these components must be tightly integrated in order to establish an efficient system.

---

[*] Presented at the First International Conference on Geospatial Information in Agriculture and Forestry, Orlando, Florida, 1-3 June 1998.
[†] This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), SF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA280-4-9044), NATO (HTECH.LG 931449), and State of Florida

Of the three components, the database subsystem is the component that can be interchanged readily by the GIS designer. Choosing a database subsystem is very important because the overall efficiency of the GIS package can be increased with the selection of a suitable database for the specific need. In today's GIS application, one frequently encounters the manipulation of a large dataset. Furthermore, simultaneous multi-user access and updating is becoming more common. This paper describes some of the database approaches available, and explains why an Object Oriented Semantic Database Management System would provide very good performance as a data server for GIS software.

2.0 DATABASE SUBSYSTEMS OF GIS

As mentioned above, the database subsystem chosen can drastically affect the performance of a GIS project, as such, the GIS designer should carefully weight the different databases available. Before choosing a database for a GIS project, the designer must consider the data storage efficiency, speed of data retrieval, user visualization of the data, data security, scalability, and many other aspects of the database.

Before we discuss the merit of different databases, it would be useful to understand the data storage requirement of GIS first. In most GIS packages, there are two separate datasets. The first set is the spatial data, which are usually represented using a non-standard data structure. These spatial data are used to store the coordinates of points, which can be combined to form arcs, polygons or other more sophisticated GIS vector data. The second set is the attribute data, used to store textual data that can correspond to different spatial data.

2.1 THE SPATIAL DATASET

ArcInfo is an extremely powerful GIS package with the capability of opening many formats of spatial data. Also, ArcInfo has also been accepted as the GIS package of choice by the industry. Thus, we have adopted ArcInfo's spatial model into this paper, which is based on the notion of vectors. Other models exist, such as those derived from the raster model. However, vector data are preferred, as it is easier for the computer to "understand" the data and manipulate them more efficiently. Furthermore, vector data model represents geographic feature similar to the way maps do.

A vector can be defined as directed line, and in the computer model, it is stored as a pair of coordinates. Thus, the basic element in the spatial dataset is a point, known in the ArcInfo jargon as a *node* or a *vertex*. The nodes and vertices are then joined together to form a line, known as *arc*. The difference between nodes and vertices are that the former are points that serve as end points of an arc, while the latter are intermediate points of the arc. Notice that each arc contains two or more points, thus the arc need not be a straight line, but it can be a curve of any kind. Optionally, arcs in turn are grouped to form *polygons*, which encloses an area on the map. Finally, all the spatial data can be grouped into layers called *coverage.*

ArcInfo uses a variety of data structures to stores this spatial data. The arc attribute table (AAT) describes arcs, nodes attribute table (NAT) describes nodes, and the polygons' and points' data are described by polygon attribute table (PAT). In addition to these feature attributes data, ArcInfo also employs the Arc-Node data structure to stores the X, Y Coordinates of all the points that make up an arc. Furthermore, Tic (TIC) and Boundary (BND) tables are used to define the geographic extent of a coverage. From these basic data structure, ArcInfo can build the database it needs to create digital

maps, which would serve as the foundation of a Geographic Information System. For more information on how ArcInfo structures the spatial data, refer to (Zeiler, 1997) and (ESRI, 1994).

2.2 THE ATTRIBUTE DATASET

The second dataset used by ArcInfo is the attribute data. They are stored in the attribute table, and consist of textual data describing the geographic features in the spatial data. In contrast to the spatial data, where ArcInfo defines the structure of the data structure, the structure of the attribute dataset is in the complete control of the GIS project designer. All that is required is that a link is established between the attribute data and the spatial data. In relational databases this is accomplished by storing the key that identifies the spatial data in the attribute dataset, thus allowing a database join command to be performed. As an example, one can have the following table to store the information about soil-type and vegetation-type of a polygon.

COVER-ID	SOIL-TYPE	VEGETATION-TYPE
60	Marl	Brazilian Pepper
. . .		

In this table, the *COVER*-ID provides an identifier that can be used to find out which polygon the record is referring to, thus a join can be established between each record and the GIS polygon with the same *COVER*-ID. The other two fields are attribute datum used for description purposes. This is because spatial data alone does not describe a real world object. Attribute data must be used to compliment the spatial data so that GIS users can effectively model their area of interest.

3.0 EXTERNAL DATABASES FOR GIS

By default, the attribute data is handled by the ArcInfo database subsystem, which uses a relational database model. If the user desires, these attribute data can be stored in an external database. ArcInfo can connect to a multitude of commercial relational database, such as Oracle, Sybase, Informix, and many others. This connection allows users to connect to their existing database and relate their attribute data to the spatial data. Even for users who design a new GIS database, the ability to store the attribute data in an external database has many advantages.

Typical commercial databases such as Oracle or Sybase offer a suit of data management tools that help to maintain data integrity and accuracy (Elmasri and Navathe, 1994):

- Concurrency mechanisms-that allow different users to query the database while other parts are being updated by other users.
- Transaction-based update where either all or none of the updates are committed.
- Schema integrity is always maintained, since only the Database Administrator (DBA) can modify the schema of the database.
- Backup and recovery mechanism to restore the database into a consistent state.
- Availability of data for other programs. The data maintained by the DBMS can be accessible by a multitude of programs, such as ArcView, or custom program that display the data over the web.

These and many other advantages are offered by a typical commercial relational DBMS. Florida International University's High Performance Database Research Center, has developed an efficient Semantic DataBase (SDB) with an object-oriented framework. In addition to the above benefits, SDB

provides many other benefits that typical relational databases do not provide. Some salient features of SDB are (Rishe, 1992):

- Optimum storage space requirement
- Efficient query processing
- Indexing on all attributes of the database
- A rich collection of GUI-based visual tools to create, load, and maintain a schema, as well as to query the database.
- High level query language support (SDB queries are simpler than SQL queries)
- Unlimited precision of numerals and fractions
- All physical aspects of the representation of the information are transparent to the users. This creates a greater potential for optimization of internal representation.

The SDB provides a standard Open DataBase Connectivity (ODBC) interface; thus this project sets out to interface ArcInfo with the SDB. We will configure ArcInfo to store all the attribute data in SDB, thus providing a way for ArcInfo's user to break away from the relational database and reap the benefits of SDB.

4.0 INTEGRATION OF SDB AND GIS

As mentioned before, all GIS software depends on a database to store its data. In the case of ArcInfo, there are several ways to do this. ArcInfo defaults on using the integrated database subsystem, INFO, to manage all the attribute data. ArcInfo can be interfaced with an external database via ODBC.[‡] There is a multitude of commercial databases that supports ODBC: Oracle, Sybase, Access, etc. Once the connection is established between ArcInfo and the external database, the Arc RELATE command allow the users to join spatial and attribute data that is accessible like a single coherent database.

Using the same approach, the integration of ArcInfo with SDB is done via ODBC. Furthermore, the SDB provides a relational interface so that ODBC compliant software can query the SDB. Although SDB uses a semantic model, no special understanding is required from the ODBC client. As such, the Arc RELATE command continues to work as if the SDB was a relational database. While accessing the data stored in the SDB, ArcInfo's users will only see the same differences as migrating from INFO to other commercial database such as Oracle. In short, the integration of SDB and ArcInfo will largely be seamless and transparent to the user; thus the benefit of SDB can be obtained with little changes in work-approach.

After the connection is established, the database must be created so that the attribute data related to the GIS project, can be stored. In the current version of SDB, the SQL CREATE command is not supported via the ODBC interface. This means that the user cannot create a table in the SDB from within ArcInfo. However, this is not a serious draw back. As anyone who has created a GIS project can testify, database table creation using commands alone is not easy, to say the least. Many would rather use visual tools, instead of word-descriptions, to create database tables and establish links between them (relations). The SDB provides a user friendly, GUI based visual tool to design and implement the database tables for use in the SDB and GIS. Figure 1 shows this visual tool in action. Once the database schema is designed, the correct database in SDB can easily be created. There is no SQL or other coding involved. Overall, the integration of SDB and ArcInfo provides a better database management environment than the traditional platform.

[‡] In the Unix environment, Database Integrator is used instead.

Another advantage of SDB is the improved speed of processing queries. This is accomplished by decomposing the query into non-redundant atomic queries that can be executed concurrently. Thus, these requests take close to the minimal number of disk accesses required to retrieve the data (Rishe, 1991).

4.1 IMAGE STORAGE IN SDB

Modern database can store more than textual data. Many commercial database can store Binary Large Objects (BLOBs). The implication is that raster images, which are often used as backdrop of GIS vector data, can also be stored in the database. Current usage does not usually exploit this feature of the database, as many see the raster image just as a file instead of a collection of records that are the realm of the database domain. However, the fact is, images can be stored in the database, and there could be several advantages to this approach. First, the images will be stored in a consistent, centralized location. Second, many descriptive data about the image that is usually not stored (or stored sporadically as files where users cannot access them easily) can now be maintained by the database. For instance, if a variety of satellite and/or aerial images are used to complement the vector data of a GIS project, many descriptive data about the image would be useful in cataloging them. Information such as date of data acquisition, coordinates of the images, data provider, description of the image, plus others information can become very handy in allowing the user to judge the usefulness of an image for a particular task at hand.

It is in binary storage where SDB has a real advantage over traditional relational databases. The object-oriented framework in which SDB was designed allows a very efficient data storage and manipulation (Rishe, *et al.*, 1995). For instance, automatic compression can be applied to the image at the database level. This will reduce the storage space requirement, and the compression can be performed independent of the image format. This is advantageous because smaller files means reduced hard disk access. This, plus the optimized query from SDB, allows large images to be retrieved faster than many commercial databases. Preliminary tests at the center showed very promising results.

4.2 EXTERNAL REPRESENTATION OF SPATIAL DATA

Another aspect that is natural for a database to handle is the spatial data. To create a vector based map, the GIS software needs to obtain the coordinates of a large number of points. Furthermore, the GIS package need to know how these nodes are connected to form line segments (arcs), and how these arcs in turn form polygons. In this section, a SDB schema that can also be used to store this spatial dataset will be discussed.

Figure 1 shows the schema necessary for storing spatial data in the SDB. Categories are shown as rectangles, while relations between the categories are shown as circles. The attributes of any category can be retrieved with a double click on the corresponding rectangle (see Figure 2). The diagram makes it very easy to understand how the SDB will store the spatial data and how the data is related. The diagram shows us that POINTs are grouped to form LINEs, which in turn form ARCs, and POLYGONs. TIC-POINT and BOUNDARY-POINT are categories derived from POINT; they have two extra attributes added to them: latitude and longitude, which give them geographic dimensions and thus identify the coverage to a particular area on the globe. If different geo-coordinate is desired, they can also be easily changed using the visual Semantic Editor. For more information of SDB design, refer to (RISHE, 1992).

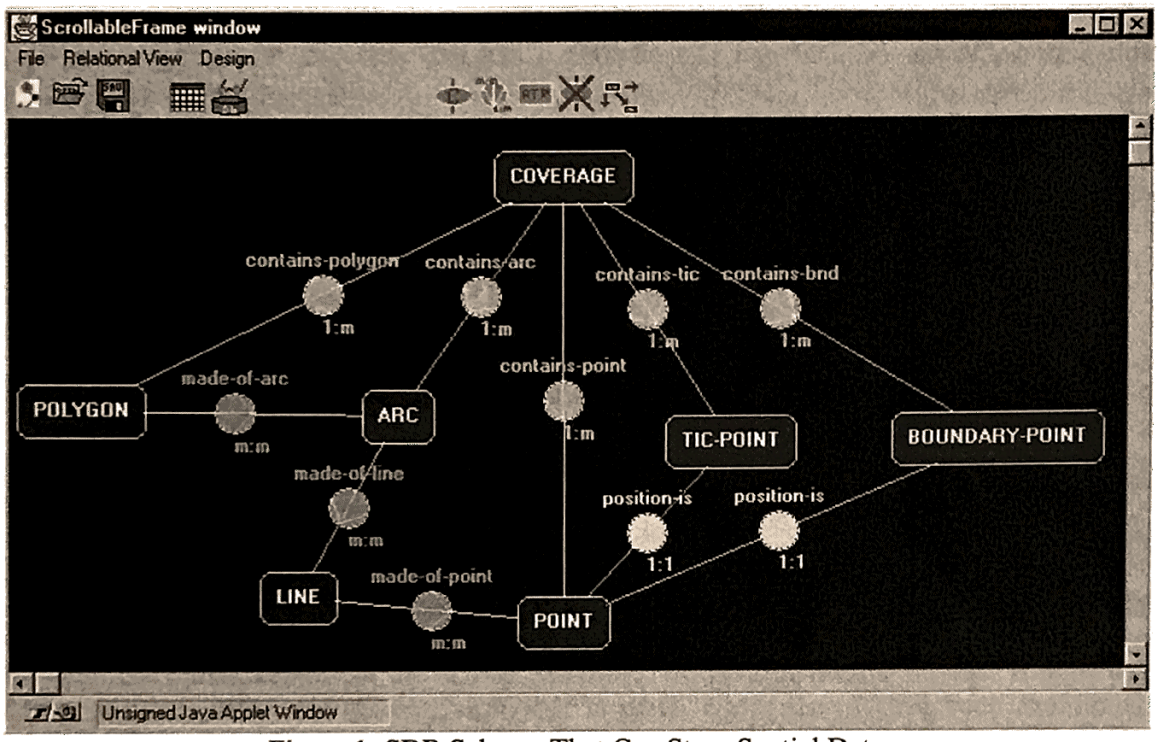

Figure 1: SDB Schema That Can Store Spatial Data.

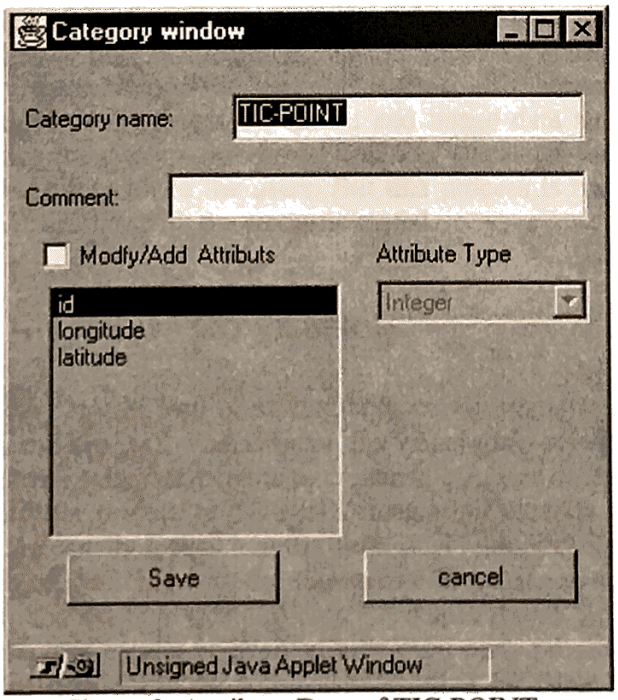

Figure 2: Attribute Data of TIC-POINT

5.0 APPLICATIONS

The integration of GIS and SDB can benefit several communities, including agriculture, environmental and forest management, which are currently using GIS for their research and studies. For example, the Database Center has a project with the Everglades National Park. One of the research projects in the park is the control of Brazilian Pepper, an unwelcomed exotic plant that has many adverse effects on the flora and fauna of the Everglades. This forestry related management part of the project calls for a database schema that requires, among other information, plot of lands that have been invaded by the Brazilian Pepper, fields where herbicides or fire has been used to control this exotic plant, as well as the effect on the land by such actions.

The experience gained by the staff in the center is that the visual database schema designer provided by the SDB is a very effective tool to create a database. The traditional relational database was hard for non-expert to understand; thus the participation level from the client was usually minimum. However, when we used the highly visual SDB schema design tool in a project with the Everglades National Park, they found it extremely easy to understand, and participated actively in the design of the database.

While this Everglades project was not born because of the GIS requirement in data analysis, it still nonetheless provided a good example of how the GIS user can benefit from the use of SDB. In fact, an extension of the current project is being evaluated, where geographic information will be added to the current text-only database, effectively converting it into a GIS project. This extension will naturally call for a modification of the current database. However, since the schema is in a highly visual and easy to understand format, such task is much easier than it had been traditionally. Furthermore, the park staff will not hesitate to perform changes, since they completely understand the project.

An additional advantage of the SDB is its flexibility. Users can use the semantic high level language to perform the queries, or they can use SQL. The semantic database browser tool allows relational-oriented users to view the schema of the semantic database as relations and attributes. Also, they can write SQL queries to retrieve all textual, spatial and image data similar to any standard relational database queries. Once the query is submitted, this tool translates the SQL query into the semantic language and provides the result to the users. But this process is transparent to the users making it very easy for users not familiar with the semantic model.

6.0 CONCLUSION

The SDB provides a very robust way of storing both conventional (textual data) and non-conventional data (spatial data and images). Furthermore, the visual tools provided by SDB provides a user-friendly interface that allows even non-experts to create an efficient database, as well as query and display the data. This current project of SDB-GIS integration extends the above-mentioned benefits to the GIS users, who are usually locked into the traditional relational database.

Still, further research is needed. Many of the standards GIS packages, such as ArcInfo, allow external databases to store only the attribute (textual) data. The spatial data structures used by these packages are proprietary and the data cannot be handled by a third party program. Thus, even when we have theorized how spatial data could be effectively handled by SDB, it has not yet been commercialized. There are three main solutions to this problem. The first solution is to program SDB to act as a middle-ware to the commercial GIS packages so that both the SDB and GIS can understand one another. A second possibility would be to create a GIS package that can interpret the SDB-based

schema mentioned in this paper. The third alternative is to wait until there is an open and standard data structure for storing the spatial data for GIS. Most believe the latter choice is more promising. Indeed, there are several organizations that are working to create this standard. The most notable organization is the Federal Geographic Data Committee (FGDC), who coordinates the development of the National Spatial Data Infrastructure (NSDI). The High Performance Research Center is following this development and would adapt the spatial data model into our SDB when a complete protocol is established.

7.0 REFERENCES

Ramez Elmasri, Shamkant B. Navathe, *Fundamentals of Database Systems, 2nd Edition,* Addison-Wesley Publishing Company, Menlo Park, CA, 1994.

ESRI and ARC/INFO, *Understanding GIS: The ARC/INFO Method, 4th Edition,* ESRI, Inc, New York, NY, 1997.

ESRI, *ARC/INFO data management,* ESRI, Inc, New York, NY, 1994.

Naphtali Rishe, "A File Structure for Semantic Databases," *Information Systems*, Vol. 16, No. 4, pp. 375-385, 1991.

Naphtali Rishe, *Database design: The semantic modeling approach,* McGraw-Hill, Inc, New York, NY, 1992.

N. Rishe, W. Sun, D. Barton, Y. Deng, C. Orji, M. Alexopoulos, L. Loureiro, C. Ordonez, M. Sanchez, A. Shaposhnikov, "Florida International University High Performance Database Research Center," *SIGMOD Record*, Vol. 24, No. 3, pp. 71-76, 1995.

Michael Zeiler, *Inside ARC/INFO, 2nd Edition,* On Word Press, Santa Fe, NM, 1997.