SEMANTIC DATABASE FOR GEOGRAPHIC INFORMATION SYSTEMS[1]

Mario R. Sanchez, David Barton and Naphtali D. Rishe
High-Performance Database Research Center
School of Computer Science
Florida International University
University Park
Miami, Florida 33199
{msanch03, barton, rishen}@fiu.edu

ABSTRACT

A practical Geographic Information System necessitates the manipulation of an extensive amount of spatially referenced data. From the retrieval process to the display function, the burden on both the hardware and software of the target system to process the data in a time restrictive manner is considerable and creates the impetus for the development of new techniques in data handling. The methods must ensure that the representation to the user is believable, useful, versatile and accomplishes the understood tasks of a GIS: to represent graphically or pictorially, either statically or dynamically, in a determinate span of time, what would be exhausting if not impossible to accomplish textually. A key facet in the achievement of these goals is how data are represented on a virtual storage medium and the manner in which these data are retrieved. The systematic coordination between the data access method and the application that processes a highly detailed visual representation must be time-tolerant for the user when the data are time relative. The challenge is to implement these ideals with the technology most prevalent to current potential users (research facilities, libraries, municipalities), with serious consideration towards all users with a desktop computer.

1. INTRODUCTION

We are developing a Semantic/spatial Information System (SemSIS) which can access vast amounts of spatial data, retrieve the data in an efficient and time critical manner, and display its representation in an adjustable real-time. SemSIS is comprised of an Application/GIS, Standard Interface, Query Language and Database Manager. This paper and related presentation is a conceptual description of the facets that make up SemSIS.

SemSIS will be comprised of four independent but highly correlated facets, each providing the features of modularity enabling their independent implementation. The Application/GIS is the module that interfaces with the user. It provides for the displaying of data and for the accepting of run-time parameters from the user. The purpose of developing an Application/GIS as part of this project is predominantly to model our advanced techniques of data storage and retrieval.

The Standard Interface module provides the bridge between the application and the query language. Allowing the application to be independent of the access methods enables the building of applications suited for a specific domain with distinct methods.

The Query Language module is the means by which to access the database. In an SQL style, it is developed for access to a semantic database and for the movement of vast amounts of data in a determinably small period of time.

The Database Manager is responsible for representing and storing massive amounts of semantic and spatial data.

The long-range goal is to apply SemSIS where the amount of spatial data is in the range from 2 terabytes to 10 petabytes and where the real-time graphical representation of such data changes continually. In one demonstration application, we are developing an animation of the time change of the ozone hole by querying TOMS [Total Ozone Mapping Spectrometer] data over 15 years.

In all cases, we are proposing to develop the mechanics to store, retrieve and rapidly manipulate vast amounts of data that represent multiple elements of a single observable source.

2. ANCILLARY MODULES

[1]This research was supported in part by NASA (under grant NAGW-4080), ARO (under BMDO grant DAAH04-0024), NATO (under grant HTECH.LG-931449), NSF (under grant CDA-9313624 for CATE Lab), and State of Florida.

0-7803-3088-9/96/$4.00 © 1996 IEEE

2.1 APPLICATION/GIS

The Application/GIS is the user interface to the system and is comprised of three basic elements.

The Standard Interface function will be the means by which the user accesses the system. All aspects of the graphical representation will be obtained from the user to be used as parameters for the retrieval and manipulation of the data. As an example, the user can enter the range of the display (e.g., the entire globe), which parameters to use, the range of dates and values, the duration of the display for range of data selected, and all queries of the database. Real-time scaling and on-the-fly changes to pictorial attributes are features being considered for implementation.

The Display function will map the data over the observed source. In essence, it will graphically display the data in real-time over a depiction, if so desired by the user, from where the data originated. The amount of time for the data to display is determined by the user. As an example, the data on ozone captured by TOMS over a period of 15 years could be mapped over the representation of the globe in a period of 90 minutes. Extending the example to any domain would enable scientists to view changes and developments (in a practical duration) that were observed over a prolonged period of time. Considerations are being made for representing the data over a virtual three-dimensional depiction of the source, as would be required for displaying ocean temperatures at varying depths.

The Retrieval function serves as the means by which the application makes the request to the database via the Standard Interface. From the parameters requested by the user, to the needs of the application to satisfy the request, all demands to the database are handled by the retrieval function. Depending on such variables as the time variant, the graphics required and the nature of the query posed by the user, the Retrieval will make the appropriate request to the Standard Interface.

2.2 STANDARD INTERFACE

All means to access the database will be made through this module. We believe that our methods for storing, structuring and retrieving information from the database is highly efficient and practical for the amounts of data that need to be handled. Requiring existing and future GIS to implement as part of their inherent code our database methods would be impractical. Although our design and implementation in this regard is an open architecture, providing a common means to access the database construct and the related language is more feasible. In essence, developers can program a finite number of calls to the Standard Interface into their application, and these calls would be the extent of the coding needed to access our high performance semantic database.

2.3 QUERY LANGUAGE

SQL's are generally designed for ordinary relational databases. A relational database construct does not easily lend itself to storing and retrieving in a time-relative manner the data size required by a high performance GIS accessing extensive databases. Additionally, the type of queries that will provide for a high degree of useability for GIS's using massive amounts of spatial data will work in the time allowed by using a semantic database [1].

As an example, consider a query of the TOMS data where the user requires all of the ozone readings for one day. On both a relational database and a semantic database, this query is relatively trivial. But now consider a query where the user wants to know the days when there was a certain reading. Or a query where the user wants to know at what latitude OR longitude a certain day produced a certain reading. In essence, consider any query in any combination of any data element. In a relational database, these combinations of queries (2^N, where N is the number of distinct elements) would be impractical if not impossible to handle in a definite and potentially short amount of time even if the relations were constructed from the onset.

Hence, Query Language will serve to handle any query to the semantic database. In preliminary tests, we have shown that the query language can make 150,000 random queries of a semantic database in approximately thirty seconds. This is the kind of performance that is required to support the Application/GIS described, or any application that requires a vast amount of related data in potentially short periods of time.

3. DATABASE MANAGER

The data will be organized as a semantic database [2]. There are several means by which the spatial-data required by a GIS can be represented and each manner is being given due consideration. First, we must analyze the amount of data that has to be dealt with.

As an example, the TOMS satellite captured 2GB of atmospheric ozone data over a period of 15 years using eight on-board sensors. In storing this data, we find a value for:

Latitude,
Longitude,
time,
reading_sensor1 ... reading_sensor8.

In essence, there are eleven values of ozone readings for each moment. Although a scientist may only be concerned with some sensor readings, we need to be prepared to deliver all eight. Assuming that a scientist wants to view the 15 years of ozone in 10 minutes, then we need to deliver approximately 3.3MB of data (that comprises the values of eleven elements) per second. A problematic but resolvable extension to this scenario is the distinctive observations by each sensor for the same space at the same time.

Consider a more prevalent requirement. A 2 terabyte data base contains the observations of ocean temperature. In storing this data, we now find a value for:

Latitude,
Longitude,
depth,
time,
temperature,
reading_sensor1 ... reading_sensorN

As can be seen with this scenario, more values, that is, a large database representing a discrete time range, the problem of data retrieval becomes a considerable concern. If our GIS was to model these data, say using variant hues for the temperature, over a specified period of time, then for our data retrieval requirement not to become an impossibility, we provide for extensions on how the database schema could be constructed. If we now extend this scenario to a 1 petabyte database, the data retrieval problem is immense, and the tacit application of a sophisticated database schema quickly becomes a dire necessity.

There are measurements that do not vary with respect to any element. For example, (ex. the temperature), may be the same for varying depths at the same or even different positions (latitude and longitude). If we take our observation, the temperature, that is the same over a three-dimensional world volume, and map it accordingly over the locations, then what we have established is a cube where the temperature is the same within the cube, and the edges of the cube are comprised of the positional measurements that define the same temperature. In our database schema, we can represent the geometry as the smallest set of non-overlapping hyper-cubes, thereby enabling the implicit retrieval of vast amounts of data with one single read. By continually containing the readings into like areas, we are using linear hyper-quadrants. A further extension is that the values in these areas need not be constant, but representable by some numerical function.

Another solution to the problem of manipulating massive data is the use of compression. There are several accepted and tested methods of data compression that we will evaluate and possibly use.

Even with the consideration of compression, the types of queries, the relation of the values of the spatial-data and the short order of time by which to move the data calls for an intricate schema on a semantic database.

4. SUMMARY

We are underway in the development of a GIS using a semantic database. One of the purposes of this approach as implemented in our SemSIS project, is to facilitate fast retrieval of vast amounts of data. Massive amounts of spatial-data are being generated from myriad sources for an indeterminate number of applications. As an example, one terabyte of data *per day* is projected to be generated by NASA's Earth Observing System (EOS) [3]. Additionally, the EOS project will permanently store several petabytes of spatial-temporal data [4]. As the amount of data to represent one piece of information in one fraction of time continues to grow, we need to represent, access and manipulate the data very efficiently and precisely. It would not suffice to tackle the problem by capitulating to the draw of infinite resources, since most scientific institutions and all future users of GIS have no such endowment.

Hence, the issues that face us are all based on moving a vast amount of data in a short period of time within the limits of existing technology. A practical consideration in this regard, is the limitation of the ethernet in moving, as an example the 3.3MB of raw TOMS data from the disk subsystem to the display - per second. We are considering the means by which we can extend our existing ethernet architecture to accommodate a true transfer rate of 100 MBPS of compressed data point to point. Other considerations in data handling are being addressed. With SemSIS we seek to provide an effective and efficient model of a GIS that will process the volume of data being generated and made available for our use.

5. REFERENCES

1. Rishe, N. *A File Structure for Semantic Databases*, Information Systems, 1991, pp. 375-385.

2. Rishe, N. *Database Design: The Semantic Modeling Approach*, McGraw-Hill, 1992, 528 pp.

3. Dozier, J. *Access to Data in NASA's Earth Observing System.* Proceedings of the International Conference of the ACM SIGMON, page 1, San Diego, CA., June 1992.

4. Short, N. et. al. *Constraint Based Scheduling for the Goddard Space Flight Center Distributed Active Archive Center's Data Archive and Distribution System.* NASA white paper, 1995.