Paper presented at the 22nd Asian Conference on Remote Sensing, 5 - 9 November 2001, Singapore.

Data Warehouse: Different Alternatives

Ajay Magon
Vice President (Technology)
ajaym@riskinc.com

Pavitra Aggarwal
Project Manager
pavitraa@riskinc.com

RMSI
A-7 Sector - 16
Noida-201 301, UP
India
Tel: (91)-118-451-1102-2101
Fax: (91)-118-451-1109, 0963

KEY WORDS: GIS data, storage, retrieval

ABSTRACT

It is widely known that the most costly and time-consuming element of a geographic analysis is acquiring and maintaining the data. The Remote Sensing workplace and trade is evolving rapidly and so is the need to organize, access and interpret data effectively, accurately and efficiently. The easiest way to store data has been a flat file. But is it the only solution? Maybe not. The demand for organization wide access to data, the requirement for solutions that provide the capability to store and manage complex spatial structures, integration with core organizational data, support large numbers of users, has driven the requirement to create database management system (DBMS) based repositories for the data.

In the spatial environment it is not only the size of data but variety of data that needs to be handled. To add to all this there are all kind of tools through which people want to access this data. Further more, people want to process this data in different ways for different requirements. These diverse requirements often require different sets of data varying in detail and quality. The complexity is further increased because of different versions in different formats for different requirements of the same data. Can there be a simple solution to this complex requirement?

This paper makes a move towards comparing DBMS based approach and file-based approach and goes on to discuss advantages and disadvantages of both. It tries to quell the debate as to which approach is better, the flat file based data storage or the DBMS or is their a third alternative available? The paper tries to build a road map towards simpler altemative to data warehousing.

INTRODUCING DATA

The foundation stone of all the sciences is data and we all know the significance of a well-structured, properly documented and easily available datasets. Companies and organizations spend lot of time and money in acquiring this data, but acquiring data is not enough. We are often challenged by the problems of organizing this data and we are often undecided as to how this data will be stored, accessed and managed. Confused and unsure we often choose solutions that are unproductive and of little value

This is true even in the GIS and Remote sensing workplace. And to add to this is the sheer size, variety and formats of GIS data. Data management becomes a complex and bewildering task. Various techniques and technologies are being used to handle this complexity, but which is the right approach is often a question that remains unanswered. Before we attempt to analyze these approaches and techniques, let us understand the data we are talking about?

GIS data, in simple words can be categorized three categories, Vector Data, Image Data and Meta Data.

1. Vector data is basically points, lines and polygons representing location and shapes of real world entities like buildings, roads, localities etc.
2. Image data includes images received from satellite, aerial photography etc
3. Meta data is description which user associates with actual data like population of area, information about customer etc.

All the above categories of data are of different type and are stored in different formats, along with different details. To add to this is the resolution at which data has been captured and the projection system, which has been used to project the data. And finally all the data will represent some geographic location, which will most likely be stored in different datasets. Most of these datasets could be overlapping or with gaps. To combine datasets of different resolution and different projections is a difficult and sometimes an impossible task.

THE TECHNIQUES

From beginning, most of GIS users have been accustomed to using flat files for data storage in various formats. This approach though simple and in use for a long time has its pitfalls in form of maintenance of multiple versions and integration with non-GIS data. The pitfalls led to users seeking other alternatives. With the increase in use of GIS data and its increased complexities RDBMS vendors came out with relational database management systems with Spatial extension, to store GIS data. ESRI introduced ArcSDE as a tool to manage and maintain GIS data allowing you to use any kind of database for data storage. Choosing the right way is thus a difficult task.

The sheer advantage of any technique can be best put forth and visualized by explaining how the technique can be applied in real world. We too, have debated our case by envisaging very familiar and common requirements using flat files, RDBMS and ArcSDE as alternatives.

Requirement 1

We begin with a commonplace requirement, which will illustrate how multiple versions of a dataset can exist simultaneously.

We have GIS data, which consists of aerial images taken from a satellite. It often happens that two or more users are using the same data for different reasons and have a need to update it. So we have a dataset version with the two users editing it for their requirements. What could be the consequences if the data is stored in flat files or RDBMS or ARCSDE? Let us go through the events that occur.

The two users unaware of each other access the same dataset and apparently do some modifications. A common situation, which often leads to complexities.

As the first user has edited the data, the version changes and a new version has been added. Similarly the second user adds another new version. How are these multiple versions going to be handled by the flat files, RDBMS and ARCSDE? The following table depicts the different approaches used by them.

Flat Files	RDBMS	ARCSDE
Since the flat files are being used, this situation creates three separate files as the files are loaded and saved separately by the two users. Thus three versions of the data is created. We have the original data as the base with two branches compromising of the two new versions created by the two users. If this has to be avoided it can only be done through tedious programming.	The RDBMS creates three records to store the original version and the two new versions. However the last record is the current and accessible. Apparently this can be handled by locking of the record being updated but this has to be done programmatically and will restrict only one user to work on one dataset at any given time.	Here user can select his area of interest from the dataset and only that part will be locked thus allowing second user to access and edit other areas from the same dataset at any given time. After the first user completes the editing, his changes are integrated with original version. Now once the second user completes his editing the changes integrates with base version.

Requirement 2

The second requirement is again inevitable, arising from the creation of different versions either while using flat files or RDBMS.

So again we have the same two users accessing the dataset now modified and versioned.

Flat Files	RDBMS	ArcSDE
Using flat files the users can either access the versions manually or programmatically.	The versions can be accessed only programmatically. Queries have to be written to access the versions	No programming is required as ARCSDE provides the users with tools to access the different versions

Requirement 3

The third requirement could be taken as a situation when two datasets in different projections need to be combined into one dataset in a different projection.

Flat Files	RDBMS	ArcSDE
Using flat files creates a complex situation. The first dataset has to be converted to a new flat file in the new projection. Then the second dataset has to be converted into a new flat file in the new projection and only then the two new flat files can be combined. This thus involves a complex process along with keeping track of the new flat files, which need to be combined.	Now here two records in different projections need to be combined and converted to a new projection. Again a bothersome task. The first record will have to be converted to the new projection, then the second record will have to be converted to the new projection and only then the two records can be combined. This can be done only using GIS tools.	Using ARCSDE you do not have to worry. ARCSDE helps you do everything on a fly.

The requirements depicted do arise often and many more commonplace requirements could be listed like issues of security, audit trails and maintaining transactions all of which led us to seek for alternatives.

The flat files are no doubt simple in structure and easy to program. They are often a very economical answer and require no extra cost of s/w to store and retrieve the data. Most of the applications read the data from flat files and provide output in flat files after processing. Data in flat files can be easily acquired and collected.

However there is the other side to the flat files leading to complexities of management. Organization and data modeling using flat files is very difficult. We also should not forget the versioning done in case of flat files and its apparent intricate management. Flat files also are a hindrance when users want to integrate GIS and non-GIS data. Further more, flat files have the inherent problem of querying the data.

The relational databases seemed to have jumped into the fray seeing the growth in use of GIS and have merely added spatial capabilities to their databases as an extension. They do make organization and modeling of data easier. The integration of GIS and non-GIS data is no longer a problem. So relevant datasets, be they GIS or non-GIS can be stored together. We all know the ease of querying a relational database; so querying GIS data becomes easy and simple. Maintenance of data, a definite negative point with the flat files becomes easier which includes maintenance of sources of change, authorization, protection of the permanent database, generation of transactions and audit trails.

The relational databases do seem to simplify things but are they really effective? Storing spatial data in normal databases is not an easy task. You need to have spatial embedded databases to store GIS data and such embedded databases are relatively new in the market. It has not been long since they are being used, so a lot has yet to seen about their effectiveness. They are definitely a costly proposition. Further more they have no inbuilt GIS related functionalities, which could ease a lot of task that has to be done programmatically. Relational databases also require learning time to learn its use. And what about exporting data from one database to another? It is not easy.

Thus the flat files though simple easy and economical pose problems of maintenance and data query along with organization. The relational databases seem to have removed many of the disadvantages of the flat files. So do we need something else to improve data access? What about the lack of GIS specific functionality in the RDBMS?

It seems that another alternative is needed to add on to the capabilities of the RDBMS and here ArcSDE seems to be the answer. ArcSDE helps to create multi-user geo databases based on popular standard RDBMS like Oracle, SQLSerevr, Informix, DB2 etc. It provides GIS related functionalities which includes jobs like conversion/extraction of data into specific projections and formats. ArcSDE has a variety of open application programming interfaces (APIs) and supports the leading spatial standards. ArcGIS 8.1 family supports ArcSDE to store, manage, access, analyze and publish spatial data in a DBMS.

CONCLUSION

In the world today where GIS related activities have increased manifold flat files are definitely not the answer. They could however still be used for one time/small requirements. However issues which can be well taken care of by RDBMS can be solved by using RDBMS with spatial extensions but for requirements of GIS related functionalities we have to depend on external tools and applications which may or may not be RDBMS compatible. To avoid this we could use ArcSDE as a middlelayer, which takes care of GIS functionalities and can communicate with most of the RDBMS with or without spatial extensions. In this way we have database experts to handle database functionalities while GIS related functionalities are left to GIS experts. A dynamic combination indeed! We are optimistic about the alternative and yet cautious as the alternative is new, and its stability and usability are still being tested.