Design and Implementation of Attribute Database Management System in A GIS System: GeoStar

Ming Xie*, Guoqing Zhou*, Deren Li and Jianjia Gong§

*Department of Engineering Technology, Old Dominion University
Norfolk, VA 23529, USA
§The Center for GIS, Wuhan Technical University of Surveying and Mapping,
Wuha, 430070, P. R. China.

Abstract

This paper presents the design and implementation of an attribute database management system (ADMS) in a Geographic Information System (GIS) system called GeoStar, which has been developed by the center for GIS at Wuhan Technical University of Surveying and Mapping (WTUSM) in China.

The ADMS was designed into 7 functional modules: file management, feature type setup, database operation, table output, statistic analysis, database conversion and help after surveying hundreds of thousands of the various GIS user's requirements and analyzing advantages and disadvantages of many commercial GIS softwares. For data share, data conversion and data communication with other databases, we used Open-Database-Connectivity (ODBC) technology to realize it. Additionally, feature identifier (ID) is designed for connectivity between graphic data and attribute data. Finally we programmed 7 functional modules of ADMS by Visual Basic. This subsystem, associated with GeoStar, has been applied in many industrial managements for decision-making, such as, Management System of East Lake Trip Area, Wuhan City, China. The result demonstrated that developed ADMS could meet the user's demands.

I. INTRODUCTION

The center for GIS at Wuhan Technical University of Surveying and Mapping (WTUSM), China has developed a GIS system called GeoStar, scheduled to spend 5 years ranging from 1995 to 2000. This system was integrated by (1) map digitizing, map editing, mapping design, map labeling, mapping symbol design, map output, image processing, remotely sensed image classification, automatic spatial data classification; (2) attribute data input, editing, querying, spatial analysis and spatial data conversion; (3) completely seamless connectivity between graphic data and attribute data [Gong 1995, 1997].

The whole framework of designed GeoStar system is illustrated in Figure 1. It is divided into 10 subsystems: (1) input and output; (2) data collection including contours and mapping symbols; (3) image processing; (4) topographic analysis and DEM application; (5) expert system analysis toolkit; (6) object-oriented spatial data management system; (7) geographic spatial analysis; (8) user developing system; (9) attribute data management and tabular output; (10) spatial data conversion.

The developments of the other subsystems have been being developed in the Center for GIS. The earlier work was reported in Gong [1995], Gong and Li [1995], Hu and Huang [1995], Zhang [1995]. This paper describes the design and implementation of an attribute data management system (ADMS), which is one of modules of GeoStar (see shaded box in Figure 1). After surveying large number of user's requirements to ADMS and analyzing the advantages and disadvantages of marketed commercial GIS such as ARC/INFO, the ADMS was designed 7 modules: (1) file management, (2) spatial feature type setup; (3) database operation; (4) tabular output; (5) statistic analysis; (6) database conversion; and (7) help. Additionally, we used open-database-connectivity to manage attribute data in order that the data in ADMS can be shared, be conversed and be communicated with other data in other databases. Feature identifier (ID) is presented for connecting attribute data and graphic data.

II. THE PRINCIPLE OF DESIGN OF ADMS

In order to meet various users' demands, we conducted the following work in design progress.

A Survey of User's Requirements

The center for GIS of WTUSM has surveyed more than million users (ask these users to present their requirements and evaluate advantages and disadvantages of marketed GIS software), has sponsored in numerous GIS symposium, has attended GIS software

1082-4006/00/0602-170$5.00

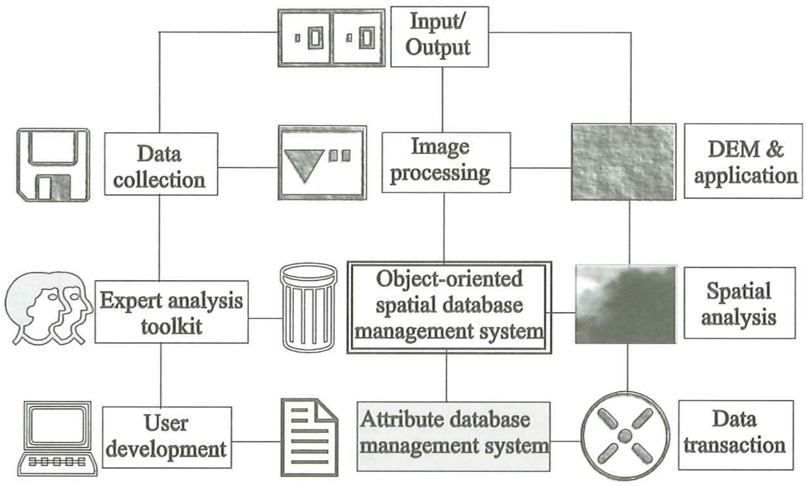

Figure 1. The architecture of GeoStar GIS.

evaluation meeting, and implements a large number of projects in China. Summarily, users in general require the ADMS to meet the following requirements:

(1) Attribute data input and management: Attribute data should be stored in the form of table. The ADMS is able to accept, process and sort various types of data automatically as well as guarantee data security always.

(2) Attribute data query: The ADMS should provide multiple query schemes in order that users can fast query things that they require in various practices.

(3) Attribute data dynamic updating: The ADMS should be designed to support different types of attribute data updating, such as, editing, erasing, deleting, adding, and so on so that the data is current, accurate and reliable.

(4) Statistics and analysis: The ADMS should have the functions of statistic analysis and development prediction to various attribute data.

(5) Data (base) operating: The ADMS should be simple to learn, convenient to operate because the users vary from beginner to sophisticated operators.

(6) User interface: Friendly user interface, such as menu bar, pull-down menu, diagonal box, pop-up menu, toolbar, and son on.

A Review to Marketed Attribute Database Management System (ADMS)

Some of commercial GIS softwares organize attribute data and graphic data in a same record of a file. This style of management is rigid and causes data redundant. Other commercial GIS softwares take attribute data as a separate data file, store attribute data with graphic data together.

Arc/Info, as a representative of many commercial GIS softwares, employed relational database to manage attribute data, while graphic data is managed like management of common files. This data model, in my opinion (Only!), exposed the following shortcomings:

(1) It only employed a partial role of database management system, and is short of concurrent control, integrity constraint, database recovery, backup, system development by user, and data model extending.

(2) Even though it makes a novel try on connectivity between graphic data and attribute data, its ADMS only is a very simple data management system in creating, operating, enumerating and managing attribute tables.

(3) Its data structure is similar to the one of dBase, but its functions are far less than the ones of dBase.

(4) It can not realize the connectivity of multiple data files through public attribute item, can not modify data file structure, and can not call data file from other directories, and has no logical type and MEMO type items.

(5) On the other hand, Arc/Info always use commends to create, delete, query, correct attribute values, display information of data, add and remove record, input and output data. These operations by commends are largely inconvenient for users [Ersi 1997a, 1997b].

Thus, to be designed ADMS in GeoStar should get over these shortcomings.

Choosing Operation Environments for Hardware and Software

Users' satisfaction is considered to be most important criteria in software development. To minimize the cost, especially in China, has become one of user's basic requirements. The low-cost, high-performance IBM–

compatible personal computer will be popular to GIS data capture, spatial query and analysis, desktop mapping, cartography. Microsoft Windows NT and Win'95 have become the most popular operating platform for current GIS software. Additionally, Windows NT and Win'95 PC can access workstations by network technology and can run almost all programs executed in Unix operating system. A statistic from GIS world shows the number of GIS software based Microsoft Windows has beyond the number based on Unix. Thus, PC-based operating platform should be considered as a priority.

The Principle of Design of ADMS

On the basis of analyses above, ADMS should have the following functions:

- Attribute data structure setup,
- Attribute data input,
- Attribute data editing, adding, deleting, and so on operations,
- Attribute data processing,
- Attribute data query,
- Statistic analysis and prediction,
- Output,
- Data security maintenance.

Therefore, the design of ADMS would keep the following principles:

Perfect: The system should be as perfect as possible. This means that the system should be able to implement many functions such as creating new file, open file, creating database, query, modifying attribute data (names, values), database conversion, mapping, output and so on.

Standard: The standardization of spatial data structure, data model and spatial data share in multiple databases has become an important issue in GIS software industry [Chen 1998]. Thus, to be developed system should conform to identical specification. Types of spatial objects should be as scientific as possible.

Advanced: The system should employ current public software and hardware environment, advanced technologies.

Compatible: Developed software can run in IBM-compatible PCs and data should be converted easily.

Effective: To be developed software should save memory and be less time-consuming; and algorithm should be optimum.

Adaptive: To be organized data can be shared and be communicated with other GIS and can be called by other GIS softwares.

High quality: inputted data is reliable, renewable, current and accurate.

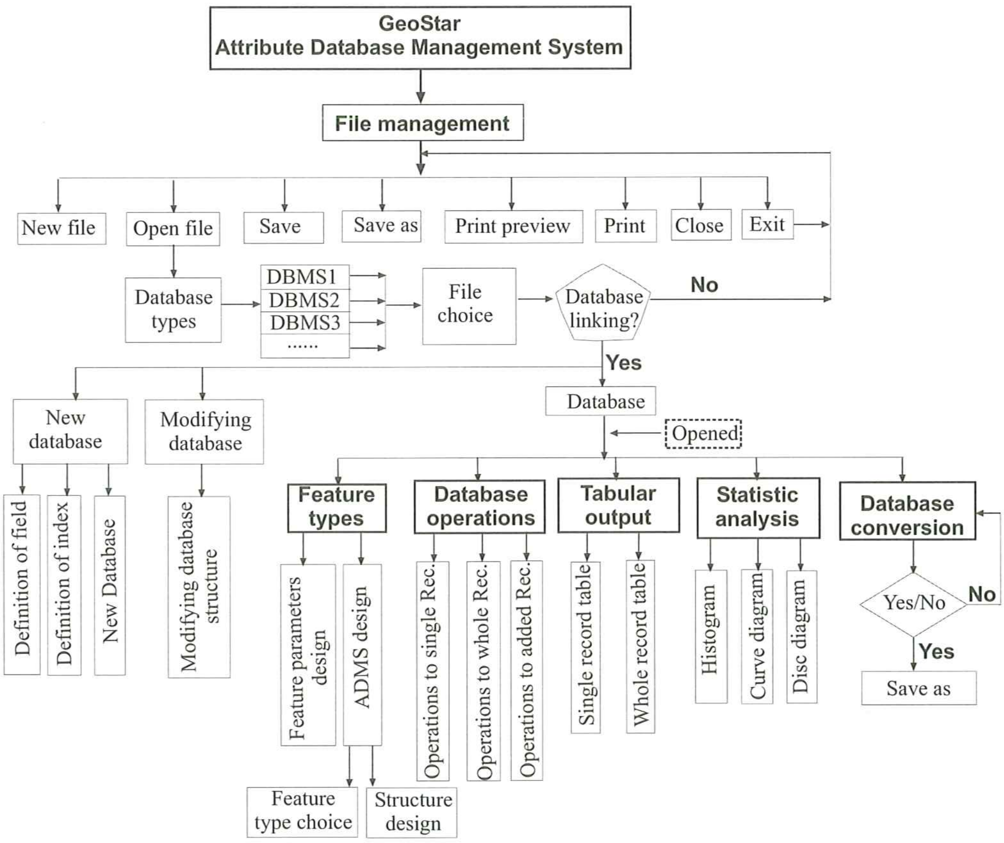

Figure 2. The designed architecture of ADMS in GeoStar.

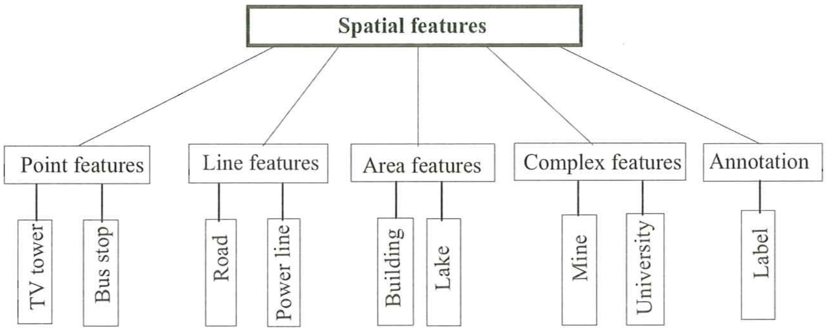

Figure 3. The spatial feature types in GeoStar [Gong 1995].

The Flowchart of ADMS in GeoStar

Figure 2 shows the designed architecture of ADMS in GeoStar, which consists of 7 modules: file management, feature type setup, database operation, tabular output, statistic analysis, database conversion and help. The advantages of this design are

(1) Open: It allows user to create client/server desktop applications.

(2) Completely seamless integration solution: Built ODBC technology and SQL connectivity serve as client/server database, which makes the GIS users access tables of relational database management system based on client/server architecture.

(3) Structure module: System was designed into tree-layer structure. Thus, each module can easily be added, deleted, moved up to users' requirements.

III. DESIGN OF ADMS

Spatial Data Types and Data Model

Before discussing ADMS, it is necessary to quickly overview the data types and data model employed in GeoStar. In GeoStar, spatial objects in the real world are thought as occurring as fence easily identifiable types: points, lines, area, complexes and annotations (see Figure 3). Line and area feature types probably consist of several arcs, and each arc has two terminal points. A feature probably has a or several labels. Thus, collectively, these five can represent most of the tangible natural and human phenomena that we encounter on an every day basis. Inside the GIS real-real objects will be represented explicitly by three of these object types [DeMers 1996].

These defined primary geographic entities are taken as basic classes, and other geographic entities are derived from these primary geographic entities. It not only inherits primary operators but define their own special operators. Thus, a spatial object can be exacted into one of feature types from down to up according to the attributes. Many feature types consist of a thematic layer. Many thematic layers in the same geographic space consist of a workspace, while a project is composed of a or several workspaces. This process of extraction from up to down and disintegration from down to up becomes GeoStar's data model (see Figure 4).

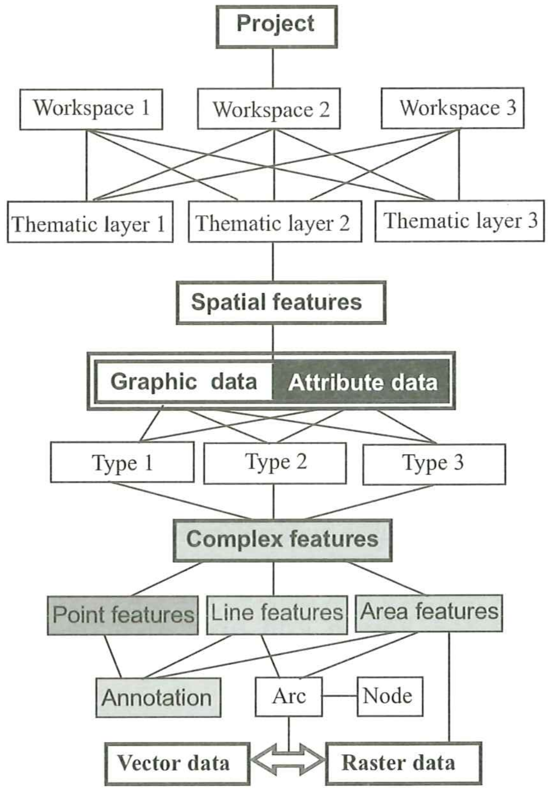

Figure 4. Spatial data model employed in GeoStar [Gong 1995]. The marked black background and white font indicates the work to be described in this paper.

Attribute Data Organization

The attribute includes feature ID, feature type, geometric values, line width, font for label, line color, fill color, and son on. The definition of spatial data type actually is to register a type of identifier (ID),

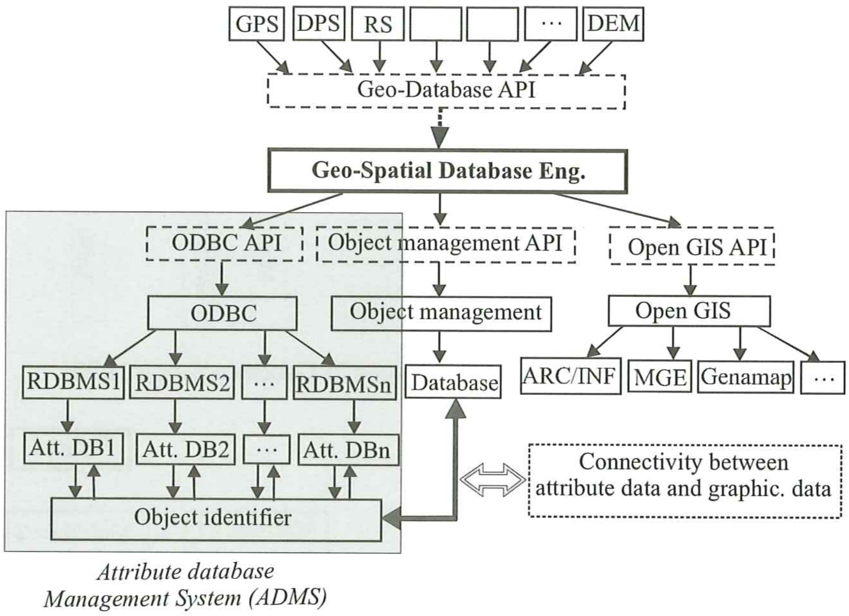

Figure 5. ODBC-based attribute data model in GeoStar.

type name, and son on. As mentioned above, the spatial data is divided into point, line, area, complex and annotation. The types of all spatial data defined by users have to be one of above types or their composition. In addition, user should indicate symbol for mapping and displaying and font for labeling such as *black, time and new roman, italic, size of font* and so on information. Fox example, a spatial object derived from linear features automatically inherit the length of item of attribute data in primary types. Similarly an area spatial object derived from area feature also inherit the length and area of primary type automatically.

After creating spatial feature, user may define the feature layer. Each layer contains many features. Each layer is actually a sub-class of all spatial feature types. The more detailed descriptions can be found in Xie (1996).

Spatial Data Management

Spatial data management system is a significantly extended relational database management system (RDBMS). The graphic data is managed by object-orientated database management (OODM), while attribute data is managed by Open Database Connectivity (ODBC). This management way provides a set of common application program interface (API), and one may ignore whatever the lower layer environment is, whichever the DBMS is. Thus this data structure easily realize database open connectivity without knowing of detailed situations of various DBMS and API (see Figure 5). In addition, this way of data management is a seamless and graded large database. It is responsible for receiving, processing, querying, scheduling, retrieving, and transmitting of spatial data and keeping the consistency and integrity of data. Moreover, it supports region locks control, region share lock, region exclusive lock and region pre-lock. Conventionally, DBMS employs different application programming interface (API). Once DBMS changes, users have to modify the original code or repeat programming so that it makes user's development and systematic maintainability difficult. Thus this kind of data model changes traditional connective way of databases.

We employed object-oriented database to manage attribute data in which spatial data sets are stored in relational database and GeoStar accesses these data by open database connectivity (ODBC). It supports various database management systems, such as SQL Server, Oracle. GeoStar supports client/server computing, large spatial database management, concurrent control in network environment, and multi-user application program interface (API). This kind of definition of spatial data model is relatively perfect because the graphic data and attribute data is a complete seamless connectivity, vector data and raster data is integrated together. The pointers to spatial features in spatial model contain not only from up to down, but also from down to up. The spatial data model enhances the adaptation and increase work efficiency largely.

The Connectivity between Attribute Data and Geometric Data

As mentioned above, some of GIS commercial software connected attribute data and graphic data by organizing them in a same record. This form of record is very rigid and causes large redundant. Other some of commercial GIS softwares take into account attribute data file as an independent file, which is stored with graphic data file together. This style of management largely restricts its application. Moreover, the data is unable be shared.

We abstract all spatial object types into point, line, area, annotation and complex features. The identical spatial object type has the same attribute item, and the attribute data is usually stored in an same database with form of tabular data. For example, if spatial object type is a point feature, we may put graphic data and attribute data together because the coordinates (x,y) of this point can directly be considered as two attribute

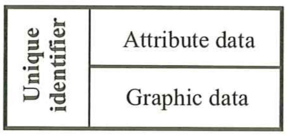

Figure 6. The connective principle between attribute data and graphic data.

items, and can be put together with other attribute values. If the spatial object types are line or area features, the coordinates are impossible taken as attribute items for the coordinates of individual point describing line and area are variable. In this case, we have to create the graphic data files for each spatial object type, and then create a unique identical code, which corresponds to each type of object, and finally create an attribute tabular data file. The connectivity between attribute data and graphic data can be carried out by unique identifier. The Figure 6 illustrates the principle of connectivity.

IV. THE IMPLEMENTATIONS OF ADMS

Visual Basic 4.0 (Chinese version) is chosen as programming language (all source codes are programmed with Visual Basic 4.0 from March 1996-March. 1997, but these codes have been changed into Visual C++ language since March 1997).

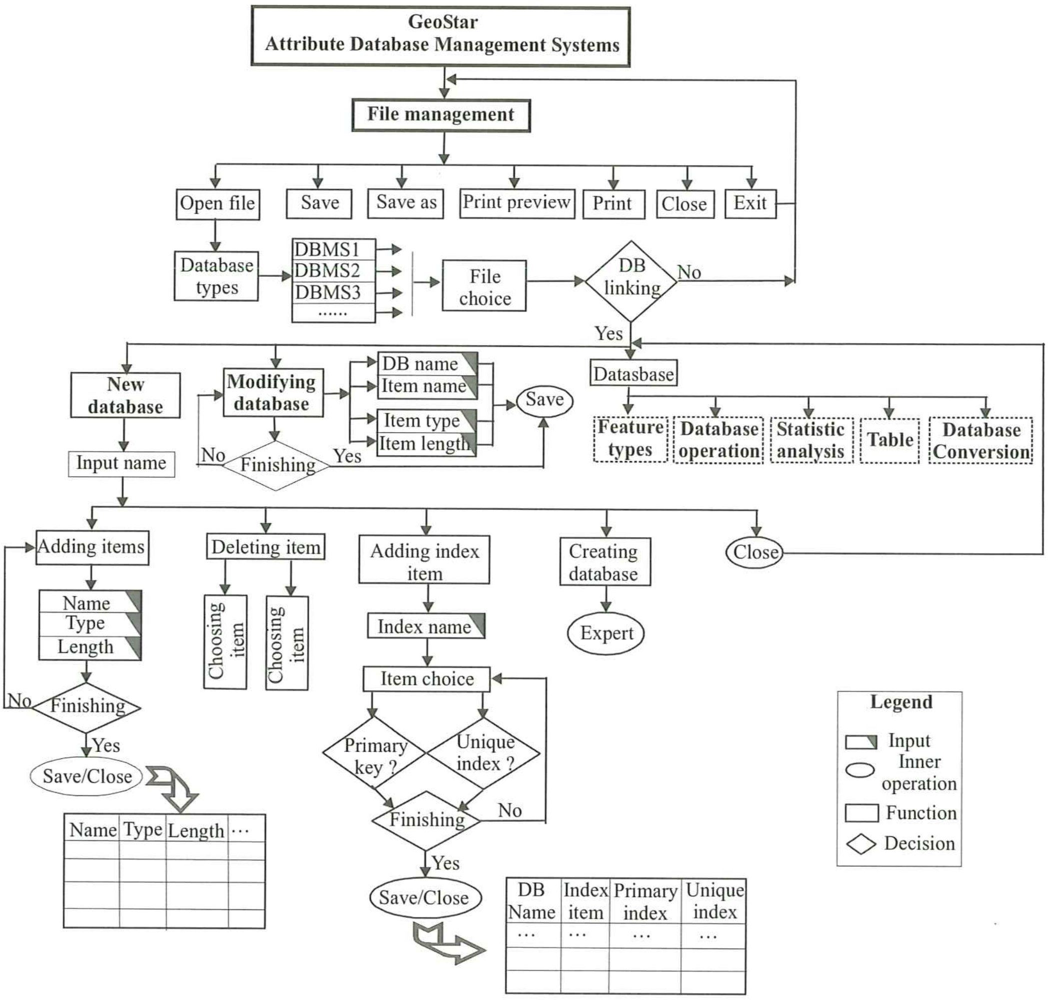

Figure 7. The whole function illustration of file management module.

The Implementation of File Management

File management module is an important phase because it has to access other modules in addition to the basic operations such as creating file, open file, save file, save as file, print preview, print file, close file and exit. Figure 7 illustrates design and implementation of this module. Representatively, here we describe the implementation of one of functions: creating file.

Creating file: If a user needs set up a new file, he first defines file name, and then defines the data structure and index.

In definition of file name, the extension of file will automatically be suffixed. Fox example, if the created database file is P3, and created database type is dBase, the extension name will be dbf. Other database types are similar. If the created database file name has existed, functional module will pop up a dialog box and prompt information: *This file has existed*, or *Whether do you cover?* or *Do you change file name?*.

In definition of data structure, user has to define attribute name, type of item, and length (byte) for each record. The data types have been designed string, double precision, integer, and so on. These data types will be displayed on screen with pull-down menu (see Figure 10. 11). User can select it. For a given variable, the length (byte) of variable will be given automatically (Table 1) .

In order to enhance the query speed in large database, it is necessary to setup an index term. The index term contains index name of database, index field, primary index and unique index.

- Index name of database indicates to define a file name in large database.
- Index field means to give a or several key items in defined file name as index item.
- Primary index means the chosen key is as key or as ancillary index
- Unique index means a unique index item.

Table 1. Variable and type in data structure.

Variable	*Byte*
Single	4
Integer	2
…	…
Double	8

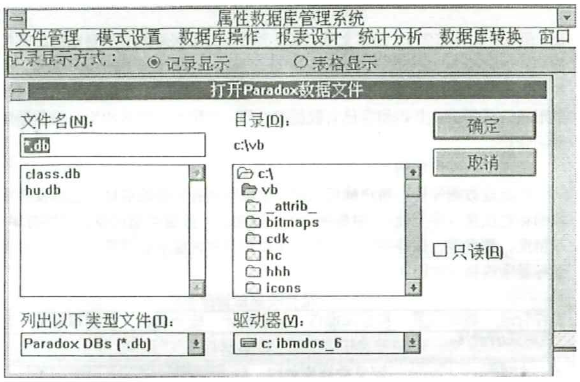

Figure 8. The file selection dialog box.

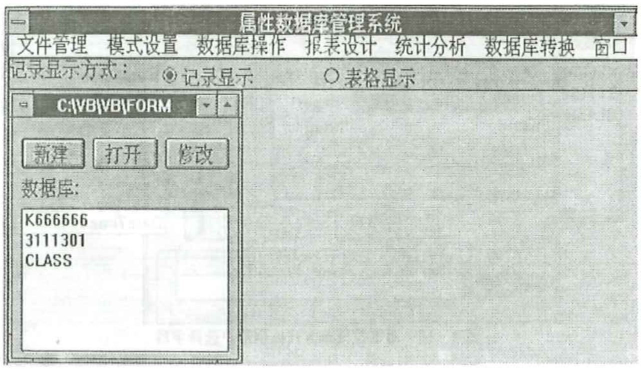

Figure 9. The interface after successfully opened file.

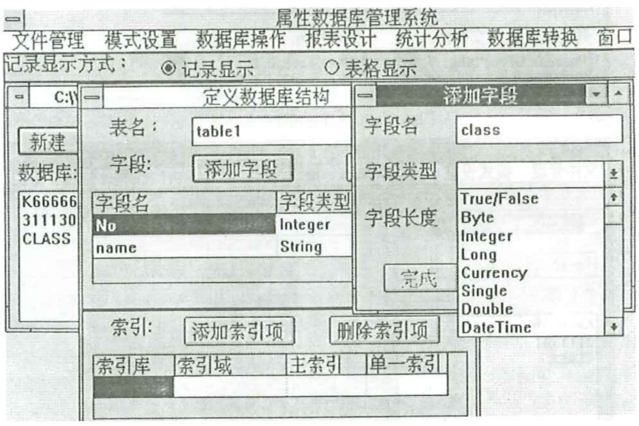

Figure 10. The dialog box for choosing field in creating file.

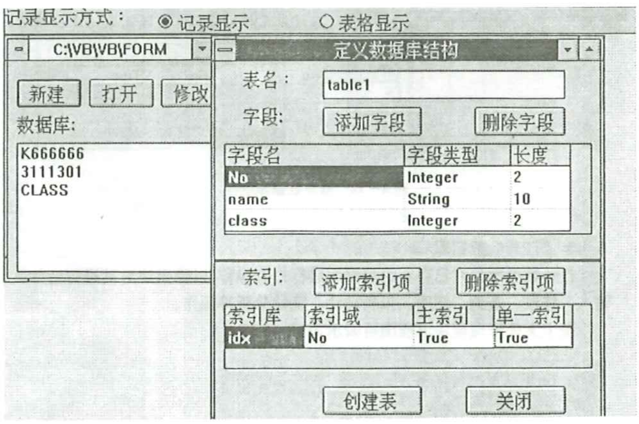

Figure 11. The dialog box for defining data structure in creating file.

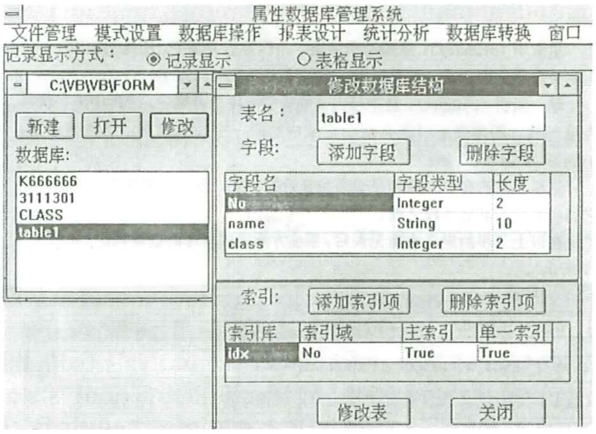

Figure 12. The dialog box for modifying data structure in the createdfile.

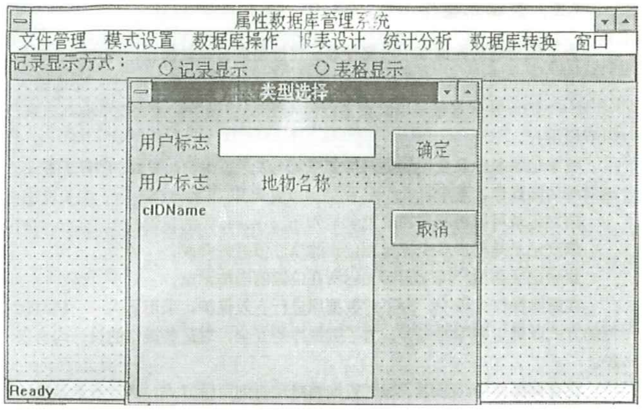

Figure 13. The dialog box for choosing feature types.

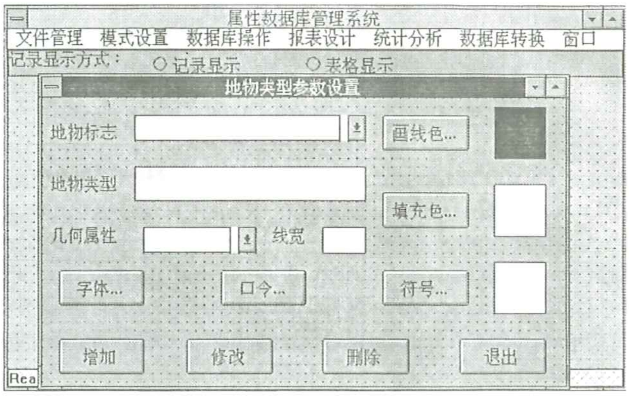

Figure 14. Choosing attribute data.

Figure 15. Data structure design in ADMS.

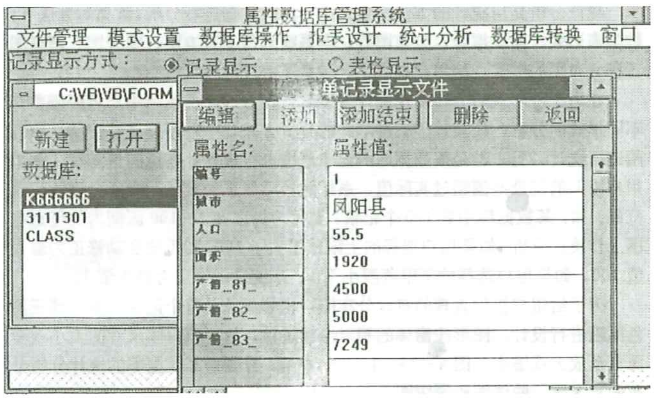

Figure 16. Data operations in single record.

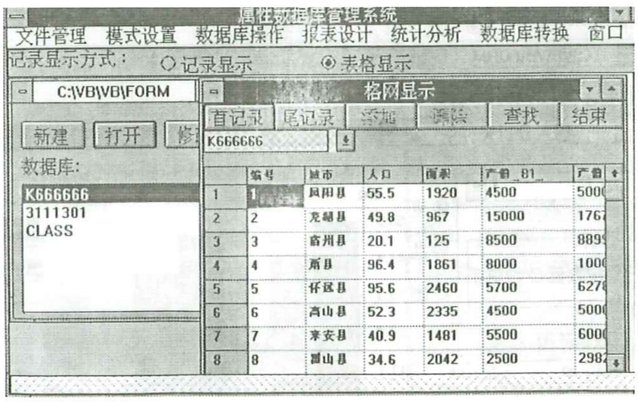

Figure 17. Data operations in whole record.

If and only if three steps are correct, a new file can be created. Figure 8 indicates an interface of file selection dialog box. Figure 9 is an interface of a successfully opened file. Figure 10 is an interface for choosing field when creating a new file. Figure 11 is an interface of defined data structure when creating new file. Figure 12 is an interface of modifying data structure.

Implementation of Data Type Setup

Data type setup includes definition of feature type and attribute data structure. Thus we programmed three dialog boxes. The first box is for choosing feature types (see Figure 13); the second box is for choosing attribute values (see Figure 14); the third box is for creating attribute data structure (see Figure 15). The types of features are described in section 3.1. The attribute values contain feature identifier, types, geometric characters, line width, symbols in mapping, line color, fill color, label font, and so on. All of attributes are list in dialog box. User only chooses it with cursor. The definition of data structure has been described in section 3.2. In the implementation of this function, the dialog boxes for input of item, type and length (byte) is designed with pull-down menu. For example, The lengths of data items, such as integer, double, single, are 8, 16, 24, bytes. User can choose them with cursor in pull-down menu (see Figure 15).

Implementation of Data Operations

Two types of operation schemes: (1) single record operation (see Figure 16) and (2) whole record operation (see Figure 17), are designed to implement this function. Both single record operation and whole record operation have the functions of editing, adding, removing, erasing, querying, and so on.

- **Adding record**: a new record may be added in any position indicated by cursor.
- **Correcting record**: user may correct attribute name, type and values positioned by cursor.
- **Deleting record**: if one of records is completely wrong, user can delete it. In deleting record progress, a diagonal box will be popped up to prompt user to make sure it.
- **Query operations**: which contain relative query and absolute query. The relative query is to search something from current position to up (backward search), or to down (forward search). Absolute query will search entire database. Additionally, the extent query (unequal equation), such as >=800 million population cities, is also programmed.

Tabular Output

Users usually need output their results of query, statistic analysis and computation in electric version (display on screen or digital file) and hardcopy version in the form of tables. Thereby we implemented two types of output of tabular data: single record table and whole record table (tabular data). Single record table in fact is a card (thus sometimes called *card output*) in which feature attribute data is described (see Table 2), and whole record table lists all attribute data (see Table 3).

Implementation of Statistic Analysis

Statistic analysis is to help user to know of statistic characters of attribute data and to predict the development. The usual statistic analysis tools have polynomial regression, multiple-factor analysis and variable analysis. So far, our statistic module is able to analyze the attribute data by histogram, curve diagram and disc diagram when data is fed by attribute value extent or record value extent. For example, if user need statistically analyze attribute values (such as ages) ranging from 50 to 100 years old, or analyze the records from 1200^{th} to 2400^{th}. The module will visualize the statistic results by histogram, curve and disc diagram. On the other hand, user occasionally wrongly input attribute values; the functional module will correct the values automatically. Fox example, if user input age ranging from -20 to 100 years old (In fact the age is always positive), this module will correct the input extend from 0 to100 automatically. Figure 18 depicts one of implemented statistic analysis: curve diagram.

Implementation of Database Conversion

This module implements the data conversion from other data formats into GeoStar data format or from GeoStar data format into others. In addition to conversion of databases each other, we still are able to implement following conversion: (1) converting all fields into other formats; (2) converting records indicated by user (for example several interesting fields) into other formats; (3) converting all items (such

Table 2. The tabular output for single record of feature attribute.

Name	Zhang Fong	**Sex**	Male	**Date of birth**	12/10/64
Title	Ass. Prof.	**Degree**	Ph.D.	**Place of birth**	Wuhan
Institution	Wuhan University				
Zip code	430070		**Phone**	27 854328	

Table 3. The tabular out for whole record of feature attribute.

Name	Sex	Title	Degree	Place of birth	Date of birth	Institution	Zip code	Phone
Zhang feng	M	Prof.	Ph.D	Beijing	04/05/1945	Wuhan Univ.	430068	8223102
Li Shi	M	Ass. Prof.	Ph.D	Wuhan	07/03/1959	WTUSM	430070	8223356
Wang Li	F	Lecture	M.D	Beijing	05/12/1961	Wuhan Univ.	430071	8224494
Zhou Qingqin	M	Engineer	Ph.D	Shanghai	11/25/1970	Center of computer	430073	8225476

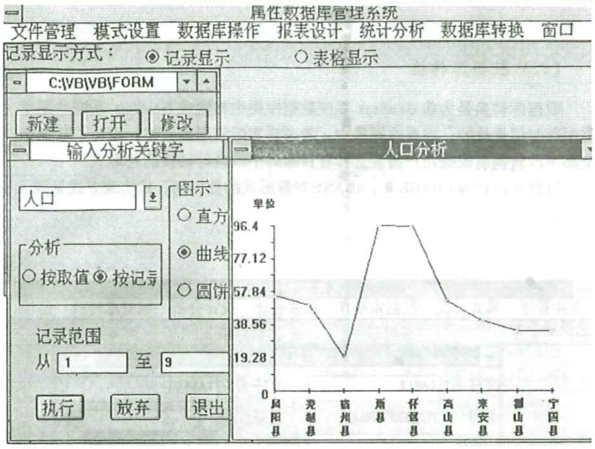

Figure 18. Statistic analysis of attr. data by curve diagram.

Figure 19. Dialog box for database conversion.

Table 4. The data structure of database.

Feature ID	Feature name	Feature pattern	Symbol	Line width	Line color	Fill Color	Font	…
1052003	Tower	Point						
2071210	Fence	Line						
3111392	Grass	Area						
…	…	…	…	…	…	…	…	…
4129029	House	Area						

Table 5. The attribute data structure of feature ID =1052003.

Workspace No	Name	Coord.			OID
1052003	Tower				

Table 6. The attribute data structure of feature ID =2071210.

Workspace No	Name				OID
2071210	Fence				

Table 7. The attribute data structure of feature ID =3111392.

Workspace No	Name	Perimeter	Area		OID
3111392	Grass				

as, attribute name) into other formats; and (4) converting interesting items into other formats. All of conversions can be carried out among dBase III, IV, Oracle 7, Paradox 3.x, Excel 7.0 so far. Figure 19 is a dialog box of implemented database conversion.

Implementation of Help

Current help pull-down menu can explains each module's function and operation in Chinese. User may obtain helps by either browsing table of contents or inputting index topics.

Implementation of Connectivity Between Attribute Data and Graphic Data

The implementation of connectivity between attribute data and graphic data has been programmed by the following procedures.

(1) All types of features are stored in a database, the data structure is illustrated in Table 4.

(2) For each type of feature, an attribute database has to be created in which various types of features have various attribute data structures. For example, the attribute data in area feature contains area, perimeter, while line feature contains length, coordinates of two terminal points.

(3) The file name of attribute database of each type of feature is taken as the identifier (ID) of corresponding type of feature. The first field in attribute database is taken as main key. If this main key contains several fields, this main key should be a compound one composed of several items. We put this main key before all fields, objective identifier (ID) at the end of whole fields. The data structures are illustrated in Table 5,6 and 7.

V. CONCLUSIONS

The developed ADMS is a sub-system of GeoStar marketed in China already. This module's function, popularity, advance and practice seriously affect GeoStar's entire function directly. Apparently the design is a key because the representation, storage, organization, management, query and update of attribute data have to be considered.

In design, we considered the fact that operating platform applied in developing GIS software can not catch up with the updating speed of computer software and hardware in input, storage and output. Thereby we adopted some technologies as advanced as possible, such as object-oriented data model, ODBC technology, visual-programming technology. Especially, ODBC technology and SQL connectivity serves as client/server database, which make the GIS users access tables of relational database management system based on client/client server. Summarily the followings are presented.

(1) To design the ADMS in GeoStar GIS software after surveying various users' requirements and analyzing the advantages and disadvantages of commercial GIS.
(2) To present using the type identifier (ID) of features + objective identifier to implement the connectivity of attribute data and graphic data.
(3) To implement all functional modules by Visual Basic programming language under the platform of Microsoft Window NT 32.

It is very hard to completely describe the more detailed implementations of each functional module in section 4. The presented descriptions in section 4 only are basic introductions of some of functions. Interested readers may index Xie (1996) and GeoStar User's guidance (1998). It should be emphasized that the ADMS described here is the first version (from March 1996-March 1997). The more powerful and updating version has been being developed.

ACKNOWLEDGEMENTS

This project was funded by National Bureau of Survey and Map in China in granted project "GeoStar: Commercial GIS Software Development". We appreciate all colleagues from the Center for GIS of WTUSM for their fruitful suggestions in design of ADMS. In programming progress, many helps from these colleagues are highly appreciated. Hundreds of thousands of annoymous users in support of ADMS user's survey are appreciated too.

REFERENCES

[1] Chen, K., 1998, *Digest: A Primer for the International GIS Standard*, Lewis publishers.
[2] DeMers, N. M., 1996, *Fundamentals of Geographic Information Systems*, John Wiley & Sons Inc.
[3] Dogac, A. and Oezsu, M. Ed., 1997, Advances in Object-Oriented Data Systems, NATO ASI Series, Series F, *Computer and Systems Sciences*, 130.
[4] ERSI, 1997, *Understanding GIS The Arc/Info Method, Self-study Workbook*, version 7.1 for Unix and Windows NT, ESRI, Redlands, CA.
[5] ERSI, 1992, *Arc/Info Use's Guide, V 6.0 GRID Command References*, ESRI, Redlands, CA.
[6] Gong, J., 1995, GeoStar-A New Generation GIS, *Geoinformatics World,* 2:7-13.
[7] Gong, J. and Wang, J., 1995, GeoStar Functional Modules and Data Model, *Journal of Wuhan Technical University of Surveying and Mapping,* 20:11-17 (supplement).
[8] Gong, J. and Xia, Z., 1996, The Trends and Perspective on GIS, China, *Geoinformatics World*, 3.
[9] Gong, J. et al., 1998, The Design and Strategy of an Object -oriented GIS Software, Proceedings of the International Workshop IAIF'97, *Image Analysis and Information Fusion,* 369-374.
[10]Hu, Q. and Wuang, J., 1995, Attribute Data Organization in GeoStar, *Journal of Wuhan Technical University of Surveying and Mapping,* 20:50-56.
[11] Xie M., 1996, *Design and Implementation of Attribute Database Management System for GeoStar*, Master thesis, Wuhan Technical University of Surveying and Mapping.
[12] Zhang, W., 1995, Object-oriented Spatial Data Model, *Journal of Wuhan Technical University of Surveying and Mapping,*20:18-22.
[13]The Center for GIS,1998, Wuhan Technical University of Surveying and Mapping. *GeoStar User's Guidance*.