A Simple OODBMS and its Application to the Pictorial Information System for Photogrammetry

Qian Zeng-Bo Chen Rong-Guo

Dept. of Photogrammtry & Remote Sensing
Zhengzhou Institute of Surveying & Mapping (ZISM)

ABSTRACT

A simple OODBMS is implemented in an OOP language Borland C++. It incorporates object identity and encapsulation via data abstraction. Objects and their operations are realised by the concept of class in C++. The DDL is constructed by C++ language. The DML is formed from lots of C++ functions. The B-tree index structure is adapted for the index management of the database. The main feature of the OODBMS is that it supports object-oriented model and can manage varible-length-record (VLR) database. Based on this OODBMS, and combined with algorithms of photogrammetry, image processing and computer graphics, an object-oriented pictorial information system for photogrammetry is built.

[KEYWORDS]

OODBMS, OOP language, pictorial information system, geographic information system, photogrammetry.

1. INTRODUCTION

The application of database technology to new applied fields, such as CAD/CAM/CAC, geographic informaion system (GIS), pictorial information system (PIS) and office automation (OA), is an extremely active area of database research. Many of these new applications deal with highly-structured objects that are composed of other objects. For example, a geographic object may be a complex object composed of different types of objects such as vectors, imagery, DEM, attributes etc. The popular commercial RDBMS, such as dBASE and ORACLE, can hardly suport these new applications because they can not handle complex objects. Several experimental OODBMSs, including GemStone (D. Maier, et al. 1987), Orion (J. Banerjee, et al. 1987), VBASE (T. Andrews, et al. 1987), PDM (F. Manola, et al. 1986), IRIS (D. H. Fishman, et al. 1987), have been developed to meet the new needs. Most of them are based on an OOP language Smalltalk with some extensions. They incorporate object identity and encapsulation via data abstraction, and support complex object and inheritance.

This paper presents a simple OODBMS which is implemented in another OOP language Borland C++ (originally Turbo C++). Since most of our application programs in OPIP (R. Chen, 1991), which is an object-oriented pictorial information system for photogrammetry, are realized in C and C++ language, we surely choose C++ other than Smalltalk as the basis for OODBMS development. The main feature of this OODBMS is that it supports object-oriented model and can manage variable-length-record (VLR) database. Based on this OODBMS, and combined with algorithms of photogrammetry, image processing and computer graphics, the OPIP has been built. In fact, the OODBMS forms the kernel of OPIP. other subsystems of OPIP are operated around the OODBMS.

2. OBJECT-ORIENTED METHODOLOGY

Object-oriented methodology (OOM) is one of the most attractive research subjects in current computer software field. The basic idea of OOM is that the problem domain is segmented in a natural way in order that the model of the problem domain can be established in such a way that it is closest to the normal way of human being's thinking, and the objective information entities can be simulated from their structures and behaviors, thereby, the designed software can represent the processing of problem solving as direct as possible.

OOM consists of three parts : object-oriented requirements analysis (OORA), object-oriented design (OOD) and object-oriented implementation (OOI). OORA is used to determine the objects and their relations in the problem domain, and establish the object model of the problem. OOD is used to design the classes of objects and the inheritant relations among classes, and build the message model. OOI is used to implement the inner status and operations for each object, and realize the message transmission among objects.

3. CONSIDERATIONS ON DATA MODEL AND DATA STRUCTURE

The data model for database has been walking towards the object-oriented model from hierarchical model, network model and relational model. In hierarchical model, data are organized as tree structure which has direction and order. In network model, data are organized as directional graph structure. In relational model, data are organized as 2-D relational table which is based on relational algebra. These conventional data models are not fit for new applications, especially for engineering applications (such as CAD, GIS, PIS etc.) because they have lack of capabilities of managing and manipulating highly-structured complex objects which exist everywhere in real world. Object-oriented model has extended and improved the conventional models. It is capable of handling and simulating complex objects. The hierarchical structure, network structure and relational structure can exist in a complex object simultaniously. Object-oriented model has expansibility so that new contents can be added to the existing data model, and different types of data objects can be held and manipulated with a unified mechanism of management.

The data types in PIS include vector data, raster data and attribute data. There are two kinds of PIS systems: raster systems and vector-based systems. Both have an important place in PIS and will continue to prevail for a long time.The latter works well when real world spatial conditions can accurately be defined as lines or edges. The vector approach can obtain important topological information which is difficult to achieve with the raster model, but it is rather ineffective for performing Boolean and overlay operations on different data layers. The raster (including compressed raster encoding such as quadtree) model is more appropriate when the problem can be described as discrete samples of continuous fields. It supports image algebra operations and has powerful backing of image processing techniques. Due to no clear answer for which one is better, the combination of vector and raster structures would be more appropriate in PIS (M. Molenaar et al., 1990).

In object-oriented model, the complex objects can be composed of different types of objects which can have different data structures and can be distributed in different databases. In this way, the vctors, rasters and attributes are combined into a unified data structure -- object-oriented structure. For example, a geographic entity may be defined as a complex object as shown in Fig. 1.

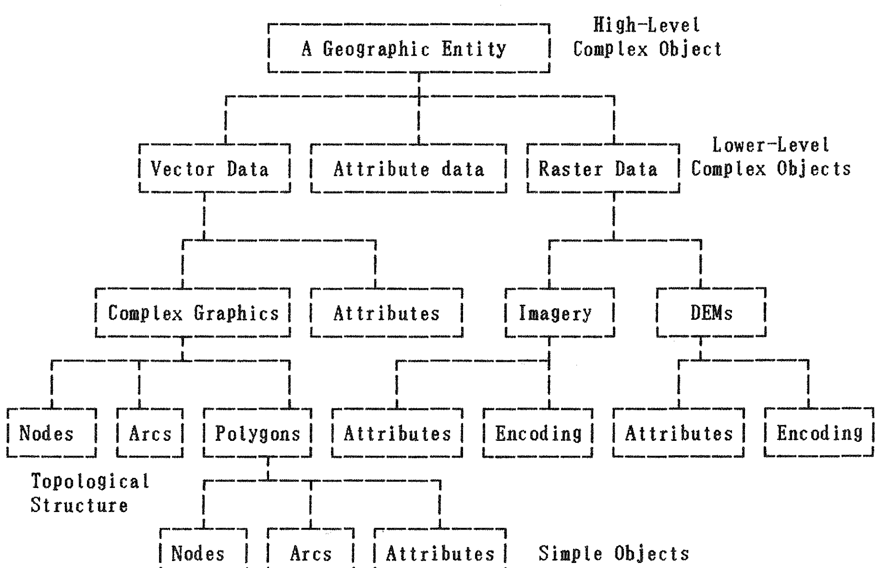

Fig.1 A geographic entity as a complex object.

4. Implementation of the OODBMS

4.1 Data description language

The objects and their operations can be implemented by the concept of class, derived class and virtual function in C++ language. The object-oriented data description language (DDL) can be realized directly by C++ language, but this kind of DDL is difficult to be maintained by the inexperienced user.Therefore, we have developed a database schema compiler. The database schema consists of three parts: data element dictionary which defines data elements and their classes, object specifications which specify the data structures and operations of the objects, index specifications which specify the index keys. The schema compiler verifies the syntax of the schema, and then translates the correct schema into C++ source codes which are used by data manipulation language and application programs.

4.2 Data manipulation language

The object-oriented data manipulation language (DML) is actually a series of C++ functions which are used to manage and manipulate databases. These functions call the data file management functions and the index file management functions directly, and then perform operations (such as addition, deletion, accession, modification, query etc.) on the databases. The B-tree structure is used for index management of the databases.

4.3 Application software

The graphics, imagery and DEMs are treated as complex objects which can be managed by the OODBMS. Considered the specific characteristics of the complex objects, we add the variable-length-record (VLR) file structure to the OODBMS. The database software is combined with image processing and graphic operations, and interfaced to the corresponding window icon system so that the user can perform various kinds of manipulations and visual queries on various types of objects (including simple objects and complex objects) in the databases.

5. Applications of the OODBMS to OPIP

OPIP is a high-tech information system which is integrated with informatics, image processing, computer graphics, database, artificial intelligence and photogrammetry. It consists of seven subsystems (as shown in Fig. 2) which are related with each other as well as relatively independent. The OPDBMS which is an applied case of OODBMS is the kernel of OPIP.

The key problem for the establishment of OPIP is to design and implement a DBMS that is capable of handling various types of complex objects (such as vector data, raster data and attribute data) and optimizing over their use, and then providing higher-level query language and visual support. Our OODBMS is designed in such a way that it meets these requirements. Based on this OODBMS, OPIP has been successfully built.

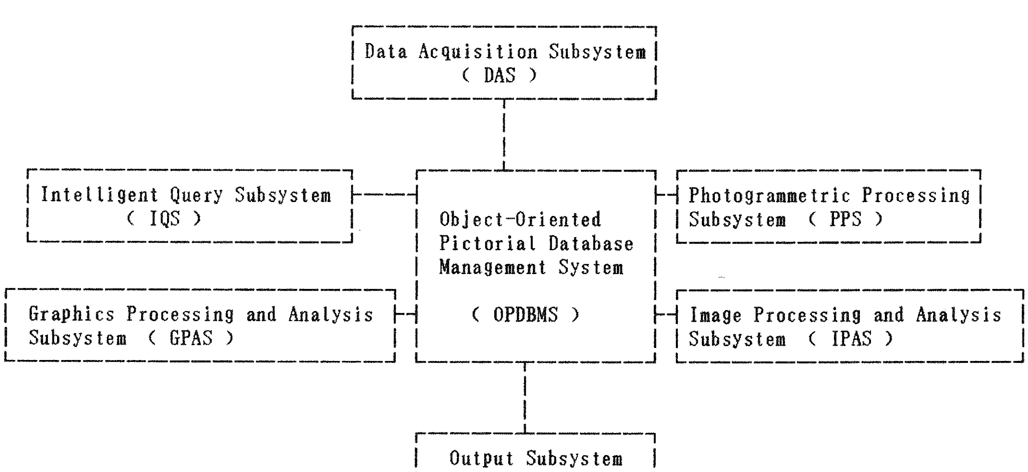

Fig. 2 Main structure of OPIP

6. CONCLUSION

The object-oriented model is a powerful model for new generation of DBMS. The concept of complex object integrates different types of objects into one comprehensive entity. The vector data, raster data and attribute data can be combined together by the object-oriented medel. A simple OODBMS which is implemented in an OOP language Borland C++ has been used to develop an object-oriented pictorial information system for photogrammetry (OPIP). Preliminary experiments show that this OODBMS has powerful extensibility and can be used for engineering applications such as CAD, GIS, PIS and OA etc.

REFERENCES

1. T. Andrews and C. harris. Combining language and Database Advances in an Object-Oriented Development Environment.ACM Conference on Object-Oriented Programming Systems, Languages and Applications. Orlando, 1987.
2. J. Banerjee, et al. Data Model Issues for Object-Oriented Applications. ACM Transactions on Office Information Systems. No. 5, 1987.
3. D. H. Fishman, et al. IRIS : An Object-Oriented Database Management System. ACM Transactions on Office Information Systems. No. 5, 1987.
4. F. Manola and U. Dayal. PDM : An Object-Oriented Data Model. International Workshop on Object-Oriented Database Systems. Pacific Grove, 1986.
5. D. Maier and J. Stein. Development and Implementation of an Object-Oriented DBMS. Research Directions in Object-Oriented Programming. MIT Press, Cambridge, 1987.
6. M. Molenaar and D. Fritsch. Combined Data Structures for Vector and Raster Representations in Geographic Information Systems. International Archives of Photogrammetry and Remote Sensing. Vol. 28-3/2, Wuhan, 1990.
7. Gong Jianya. Object-Oriented Models for Thematic Data Management in GIS. International Archives of Photogrammetry and Remote Sensing. Vol. 28-3/2, Wuhan, 1990.
8. Chen Rongguo. OPIP : An Object-Oriented Pictorial Information System for Photogrammetry. Proceedings of the Symposium of Chinese Academy of Sciences on Remote Sensing and GIS. Beijing, 1991.