DESIGN AND IMPLEMENTATION OF AN OBJECT-ORIENTED GIS SOFTWARE

Jianya Gong Deren Li

National Key Lab for Information Engineering in Surveying, Mapping and Remote Sensing
Wuhan Technical University of Surveying and Mapping
39 Luoyu Road, Wuhan, 430070, P.R. of China

Commission III/IV

KEY WORDS: GIS, Database, Software, Design, Model, Integration

ABSTRACT:

This paper discusses object-oriented technology in GIS from geographic real world, through the conceptual model of geographic phenomenon, logical model of geographic database, and software modeling of geographic information system. The object-oriented semantic models are used for describing the spatial phenomena and objects. An object-oriented spatial database management system (GeoDB) is designed according to the object-oriented logical data model, and an object-oriented GIS software (GeoStar) is developed based on the GeoDB and C++. Some techniques on object-oriented analysis, object-oriented system design, and system implementation are introduced in this article.

1. INTRODUCTION

Object-oriented technology has come into fashion in computer science and technology. It includes Object-Oriented Programming Language (OOPL), Object-Oriented System Analysis (OOA), Object-Oriented System Design (OOD), Object-Oriented Interface (OOI), Object-Oriented Database Management System (OODBMS), Object-Oriented Geographic Information System (OOGIS), Object-Oriented Knowledge Engineering (OOKE) and so on. We have seen that object-oriented approaches have made great progresses. Proprietary object-oriented systems have continued to develop in the intervening years. The programming language C++ is very popular and now well-established as a standard object-oriented language. Commercial object-oriented database management systems (OODBMS) including ONTOS, Gemstone, ObjectStore, O2 and etc. have become easily available, although they are still not in use on large-scale. In GIS, research on the application of object-oriented ideas to GIS has come a long way in the past few years, from pioneering work by Egenhofer and Frank (1987), through the explorations done by Worboys(1990), Gong(1990), etc., to current work on conceptual models, logical data models, system modeling and implementation. The object-oriented approach has been used to design and implement systems such as Ietergraph's TIGRIS, Smallworld GIS, Laser Scan's GAE and GeoStar (Gong, 1994) developed by Wuhan Technical University of Surveying and Mapping in China. As yet there are few proprietary GIS that have object-oriented features, but those that do exist have been highly successful for specialized markets and it is likely that more commercial object-oriented GIS will follow(Worboys, 1994).

A GIS is a multi-faceted entity that goes through several processes in its life-cycle (see figure 1, Worboys, 1994) from conception through design and implementation to use. The object-oriented approach can be applied at all of these different levels and has a slightly different meaning at each. From the beginning, the object-oriented conceptual model can be used for object analysis and system analysis. Following analysis comes design, which is responsible for deriving a logical data structure and a system.The procedure from object-oriented conceptual model to object-oriented logical data structure is natural and the result of object-oriented analysis is useful to object-oriented design. The translation of the design

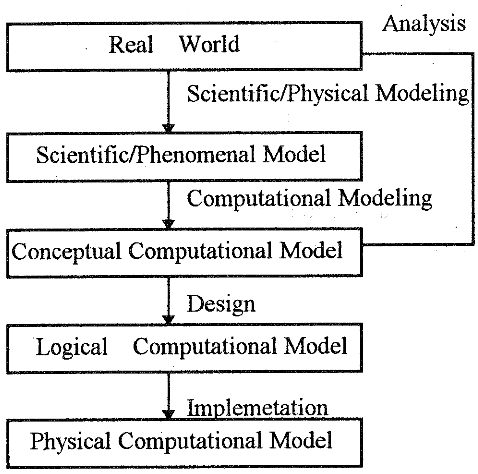

Figure 1 System life-cycle of GIS

into something that really works is called the implementation stage. In this stage, object-oriented database management systems and object-oriented programming languages are suitable mechanisms to implement the object-oriented systems. When the life-cycle of GIS moves to usage and maintenance, software reuse and inheritance techniques can be used to develop and maintain user application systems.

This paper discusses the conceptual model of geographic phenomena by object-oriented analysis in section 2; object-oriented logical spatial data model in section 3; design of object-oriented spatial database management engine(GeoDB) in section 4; software modeling implementation of an OOGIS software (GeoStar) in section 5; and finally a conclusion in section 6.

2. OBJECT-ORIENTED CONCEPTUAL MODEL IN GIS

Figure 1 shows several levels of the system life-cycle and their translation. For geographic information system, various phenomena on the earth's surface are scientifically translated to images via photogrammetry and remote sensing, or maps via cartography and field surveying. The phenomena or objects on images or maps can be represented by an object-oriented model. The following semantic concepts are important in OOGIS. They are *Object, Classification, Generalization, Aggregation, Association.*

When we look at images or maps, we can identify buildings, parks, lakes, highways, roads, railroads, rivers, power lines, towers, train stations and etc., then classify them to feature classes. Furthermore, class *building* might be specialized into classes *supermarket, post office, hotel, restaurant, house and so on*. From the other side, we can combine classes *building, park, lake and etc.* into a common superclass *surface*, because they have the same geometric properties. Several features might be aggregated to a complex object, for instance, some buildings and other features can be grouped to a university or a factory.

In geometry, there are four highest superclasses (POINT, LINE, SURFACE, and COMPLEX) abstracted from all features of GIS (see figure 2). All spatial features are defined as belonging to various classes. Each feature class is declared as belonging to a superclass of four geometry classes, which are distinguished according to representation but not phenomenon. A characteristic of such a feature class is called an attribute (non geometry). Certain attributes are identified with each class. For example, the class BUILDING has attributes like *building number, street, zip, city, owner, usage, built date, floors, etc.*. The specialized subclasses of a feature class might have additional and more detailed attributes, and can inherit the common attributes from their superclass. For example, Class HOTEL is a subclass of class BUILDING, it inherits the attributes of BUILDING like building number, street, zip, city, owner, function, built date, floor number, etc. It also has some special attributes such as employer , manager, room number, bed number, income, tax, etc. (see figure 2).

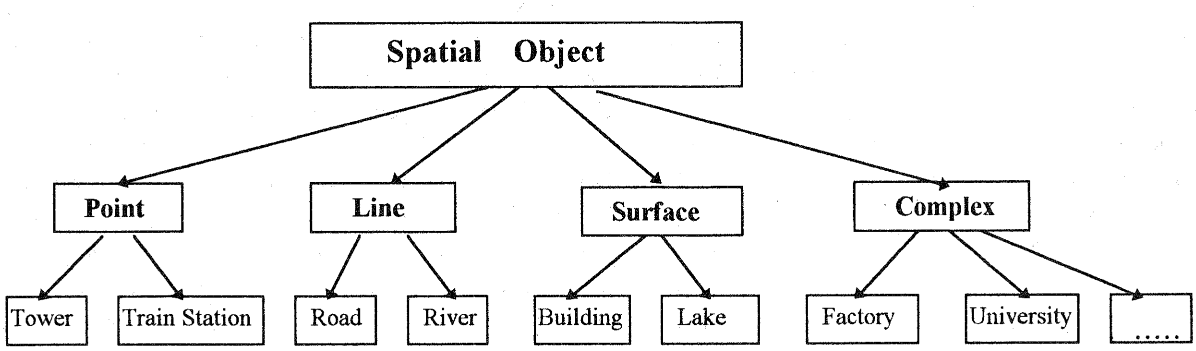

Figure 2 Feature classification

3. OBJECT-ORIENTED LOGICAL DATA MODEL IN GIS

The conceptual data model is abstracted from natural features or their images. The model needs to be translated to logical data model so that it can be represented in computer. For GIS, attribute data are usually expressed by relational data models. Similar to relational data models, object-oriented data models might be designed for attribute data in GIS, because object-oriented data model can contain and extend relational data model. Here the emphasis is on geometric data organization since it is more complicated.

A very important concept in GIS is the topological relationship between spatial objects. Two basic relations for most GIS applications are connection relation and association or aggregation relation.

Thare are five geometric classes: COMPLEX, SURFACE, LINE, NODE-POINT, and one LOCATION type. They are organizated via aggregation semantic model. An object-oriented data model in GIS is shown in figure 3.

Object-oriented data model is different from relational data model in that the former not only supports variable length attributes, but also can directly deal with SET OF OBJECTs in an attribute field. In this case, a single structure table represents a type of spatial object, which is defined as a class. Therefore, the semantics of the object-oriented data model is more natural and clearer than the relational model.

A partition should be established above the Spatial Objects. All objects belong to the partition. Several partitions are organized into a project. In order to satisfy different requirements for application, display and mapping, thematic layers can be custom-designed. Each theme might contain several feature classes. Here thematic layer is not like coverage, with the former organized only in logical, not physical. sense.

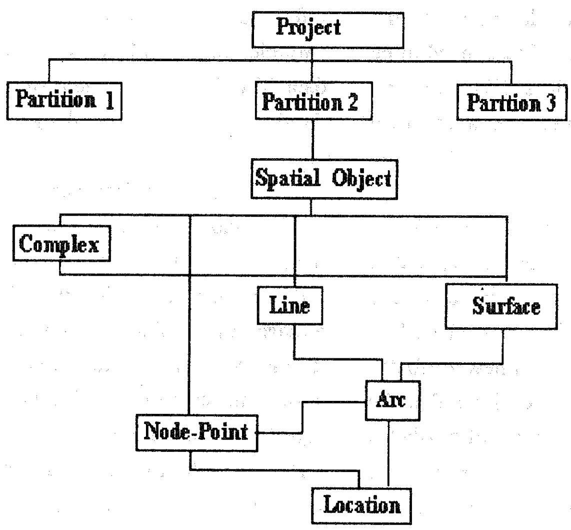

Figure 3 An object-oriented spatial data model

4. OBJECT-ORIENTED SPATIAL DATA MANAGEMENT ENGINE

An object-oriented spatial database management engine called GeoDB developed by Research Center for GIS at Wuhan Technical University of Surveying and Mapping is designed based on the above spatial data logical model. The engine is actually a class library of spatial data. The database programming language is an extended variant of C++. It uses the same compiler tools and context as C++.

The system-defined classes constitute an engine of database management. The user can model the application domain by using these classes. The system-defined classes mirror the spatial object model semantics. SPATIAL OBJECT is the root of an object lattice. The class SPATIAL OBJECT is a superclass in C++, where there is no attribute, but it includes many functions such as attribute data structure definition, insert, select, delete and update attribute records, index and join tables, and other operations like a general database management system. These operations are responsible for dealing with attribute data in GIS.

Spatial objects can be divided into point feature, line feature, surface feature and complex feature. Four corresponding classes in C++ are declared as subclasses of the class SPATIAL OBJECT. At the same time, they are supercalsses of all features in GIS (see Figure 2). Generally, the feature classes are defined by the user. Except SPATIAL OBJECT, NODE-POINT, LINE, SURFACE, and COMPLEX, the classes ARC, ANNOTATION, PARTITION, PROJECT and type LOCATION are system-defined. The class PARTITION and the class PROJECT contain some matedata attributes and spatial operations like spatial index, attribute merge in different partitions etc..

System-defined classes have system-defined geometric attributes and some operations, which are inherited by the user-defined subclasses.

Objects are identified by an object identifier (OID). An objec identifier is generated by the system and represented by the system-defined attribute OID which the user can refer to. The object identifier is not reused even when the object is deleted. Geographic information systems using object identifier have the advantage that only the topology of modified objects need to be rebuilt when spatial data are edited. While a GIS based on records must re-build topology for all objects in a coverage when an arc or a polygon is modified.

The system-defined classes containing attributes and methods, are an extended subset of C++, which form an engine for spatial database management. Actually they are functions and attributes of Application Programming Interface (API) based on C++. The user can define new classes, and add new attributes and operations for user-defined classes. If he declares that the new class is a subclass of system-defined classes, all attributes and operations from its superclasses, including superclass of superclass, for example class SPATIAL OBJECT, are inherited. The engine provides sufficient system-defined classes with their attributes and operations to make possible a wide variety of applications in GIS.

In GeoDB, spatial topological relations and complex objects are represented by object identifiers. Nested relations make it possible to represent complex objects through clustering. There are seven geometric classes in Figure 3, which are *COMPLEX, SURFACE, LINE, NODE-POINT, ANNOTATION, ARC,* and one LOCATION type. Each spatial object has an identifier, which contains class identifier, so the inspection of an object identifier of an object can determine the class to which the object belongs without looking up the content of the object. While the type LOCATION contains only coordinates not identifier. In this model, classes COMPLEX, SURFACE, LINE, ARC, and NODE-POINT are feature classes which have attributes. Class ANNOTATION is a secondary class and each annotation is related to its object.

In object-oriented logical data model, an object class can be constructed into a table, in which we can explicitly declare its relations with other classes. Unlike relational data model, a table can not represent an entity type and the relations. The object manipulation functions support associative access of object and direct access of complex object without explicit joints.

The GeoDB provides some main functions of commercial database management system like page buffer, object buffer, data security, integrity control, multiple user access and concurrency management, object index, spatial index, and etc.. The system architecture is shown in figure 6.

The data management subsystem provides nested relations and supports variable-length records so that it can efficiently store objects, which have several kinds of non-formal structures such as variable-length attributes, set of object identifiers and multi-value attributes. A tuple represents an object. When information that an object contains can not fix its length, a variable-length tuple need be set to store the variable-length data. The data management subsystem provides two data types, fixed-length and variable-length fields.

The data management subsystem in GeoDB can store both spatial data and non-spatial data. It has the same storage management functions as commercial DBMS. However, in order to interface with some other general DBMS like Oracle, Sybase, Ingres, Paradox and etc., a gateway in GeoDB will be developed as a general-purpose extended relational database system to support the object-oriented spatial data model and provide the operations on top of RDBMS. (We can adopt the ODBC as API of the gateway). It replaces the data management in GeoDB for data storage. The object management, object buffer, object index, spatial index, object manipulation, schema definition, and spatial topological relations building and maintainenance are still accomplished by the object-oriented spatial data management engine.

5. OBJECT-ORIENTED GEOGRAPHIC INFORMATION SYSTEM

An object-oriented geographic information system based upon the object-oriented spatial database management engine (GeoDB) has been developed by the Research Center for Geographic Information System at Wuhan Technical University of Surveying and Mapping in China. The system is named GeoStar. It is developed with C++ and includes the modules shown in Figure 4.

The module for spatial data capture and edit includes digitizing on tablet, vectorization from scanned maps, data editing, snap to node, topology building and error check, edge matching, input and edit attributes, and other functions.

The image processing module has the functions including radiometric corrections, sensor corrections, merge data set, geometric corrections, filtering, contrast stretch, color domain conversions, density slicing, histogram, histogram equalization, mosaic, band radios, supervised classification and unsupervised classification.

The third module is specially for desktop mapping and spatial query. It can be used to make maps that involve graph and tables of quering result. Queries from both directions are available. User can query the attributes of a spatial objects by cursor, and from the other side, user might get a tables and a set of objects by selection query.

The spatial analysis functions in GeoStar are divided into two modules. The two-dimensional analysis model is responsible for buffer analysis, network analysis, polygon overlay, point in polygon, line in polygon and other. operations. While the DTM and application module provides TIN and GRID generations and surface analysis functions such as generating perspective view, generating contours, generating line-of-sight, generating cross-sections, and generating fractal visualization surface.

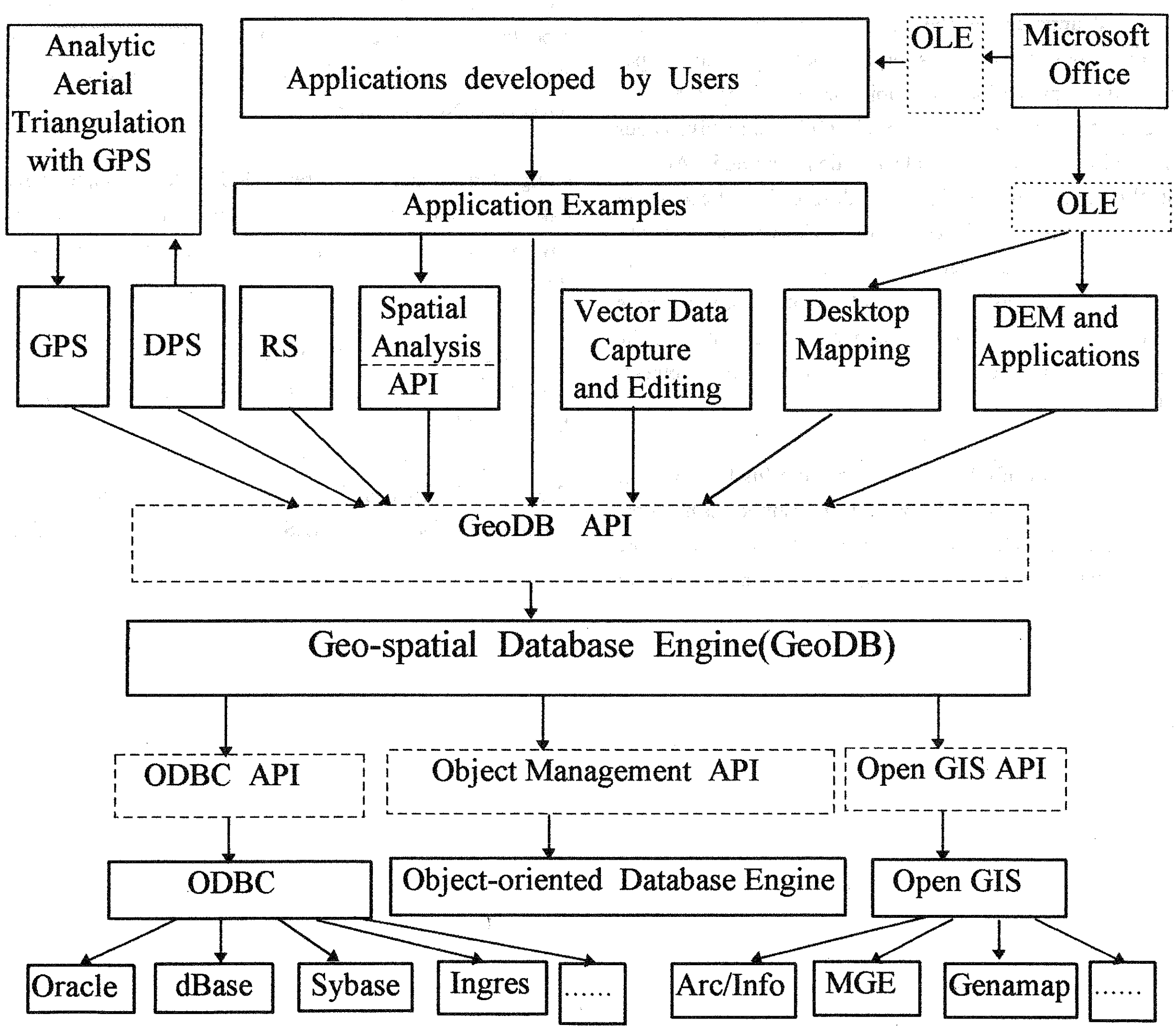

Figure 4 The modules and design framework of GeoStar

All modules share the same data models and data structure, and use a database management engine and an API library. Each subsystem has its own interface, which can be modified by the user.

We can design the spatial object classes in C++ corresponding to the spatial data model in Figure 5. Each kind of spatial object uses a C++ class, but it is a temperary class unlike the classes in an object-oriented database management engine which are permanent. Every temperary class might have the same data members or more as its permanent class, because some control variables or additional information might be included in the temperary class.

6. CONCLUSIONS

Object-oriented approach can be applied to all aspects of the life-cycle of information processing. The main advantage of object-oriented approach is that the analysis and design of a system are consistent with its object models. This paper introduces the applications of object-oriented approach at various levels through the life-cycle of a GIS. Some advantages and potentialities have been clearly demonstrated. At this moment, the major obstacle for the widespread applications of object-oriented technology in GIS is the lack of popular object-oriented database management systems. Scientists and engineers have accepted the object-oriented programming languages as a main tool. The next step is to pay attention to the object-oriented database management system and object-oriented GIS.

From the viewpoint of GIS development, the final resolutions of many problems might rely on object-oriented technology. Three dimensional GIS and temporal GISs need more complicated data models and concern more nested relations. Object-oriented data model and database management system can represent complex objects and nested relations. These might prove to be most promising for 3-D and 4-D GISs.

Reference

Chance, A., Newell, R., and Therault, D., 1990, An object-oriented GIS: Issues and solutions. In Proceedings of European Geographical Information Systems(EGIS)Annual Conference. pp. 179-188.

Egenhofer, M. J., and Frank, A., 1987 Object-oriented database: database requirements for GIS. In Proceedings of the International GIS Symposium: The Research Agenda, Vol.2, pp. 189-211.

Gong, Jianya and Li, Deren, 1992, Object-oriented models for thematic data management in a GIS. *Australian Journal of Geodesy, Photogrammetry and Surveying,* No. 56, pp. 37-48.

Gong, Jianya and Li, Deren, 1992, An object-oriented data model based on the unified data structure. *Archives of 17th ISPRS Congress*, Washington, D.C., pp.772-779.

Ishikawa, Hiroshi, 1993, Object-oriented database system -- design and implementation for advanced applications, Springer-Verlag Publishing House

Worboys, F. Michael, 1994, Object-oriented approaches to Geo-referenced information, IJGIS, Vol.8, No. 4, pp. 385-399.