

**EGG : A GEOGRAPHIC INFORMATION SYSTEM
USING OBJECT-ORIENTED CONCEPTS[†]**

Byoung-Woo Oh, Ki-Joon Han

Department of Computer Science, Kon-Kuk University

93-1, Mojin-Dong, Kwangjin-Gu,

Seoul 133-701, Korea

e-mail : {bwoh, kjhan}@galaxy.konkuk.ac.kr

Abstract

In this paper, we design and implement a geographic information system using object-oriented concepts, named EGG (EXODUS-GOOD-GRACE), that consists of the User Interface layer, the Data Manager layer, and the Storage Manager layer. The User Interface layer (GRACE : Graphic User Interface for GIS) allows the administrator to manage geographic data easily and permits the end users to display the results clearly, the Storage Manager layer (EXODUS Storage Manager) stores and retrieves a lot of geographical data efficiently, and the Data Manager layer (GOOD : Geo-Object Oriented Data Manager) connects the two layers. Since the query language approach is not suitable for an API of GOOD, we implement GOOD as a library of callable functions. In order to develop the EGG geographic information system, we first analyze each of the layers well enough and then implement the individual layers by using suitable development tools. Lastly, we develop the complete EGG geographic information system by combining the three layers. Since the layered approach provides transparency and reusability, it can reduce time and efforts in developing future GIS applications.

1. Introduction

In diverse applications, information systems are being used increasingly as a means to manage, retrieve, and store large quantities of data which are difficult to handle manually. Spatial data handling is no exception. It is called GIS (Geographic Information System) that stores and retrieves geographic data, and analyzes and processes them to supply the useful information for geographic applications. It is very difficult for application programs alone to handle plenty of geographic data efficiently. Therefore, GISs use a database management system for handling the geographic data [4, 8, 10, 12].

There exist several approaches in implementing GIS using database management systems. First, the use of a relational database management system has been a general approach until recently. In this approach, however, storage management is often handled by two separate systems: the relational database management system manipulates alphanumeric data and the other system handles spatial data. One example is the ARC/INFO commercial system [11] which is divided into two parts. This system uses a relational database management system to manipulate aspatial data and a file system to handle spatial data. This approach has some disadvantages caused from the file system.

Second, an object-oriented database management system is used to manage spatial and aspatial data together. One example is the GODOT geo-object management system which uses the commercial object-oriented database management system, called ObjectStore [7]. However, lack of reliable object-oriented database management systems and the nonexistence of a standard for object-oriented database management systems make this approach infeasible.

Third, the use of a development tool to build a database management system for implementing GIS is now considered to be the most desirable approach [12]. This approach can provide the efficient manipulation of both spatial and aspatial data, but it needs more implementation efforts. In this paper, we adopt the third approach and use the EXODUS Storage Manager [3] that is a powerful tool for implementing an object-oriented database management system. In addition, our GIS uses object-oriented concepts to model geographic data.

In this paper, EGG (EXODUS-GOOD-GRACE) GIS is layered into three parts: User Interface layer, Data Manager layer, and Storage Manager layer. The User Interface layer that is an application layer is divided into Class Manager, Object Manager, and Query Manager. The Data Manager layer that provides the efficient management of geographic data is divided into data manipulation module, image manipulation module, and import/export manipulation module. As a Storage Manager, we adopt the EXODUS

[†]This research was partially supported by the Korea Telecom under No. 94-02, and partially by the Korea Science & Engineering Foundation (KOSEF) under No. 941-0900-079-2.

Storage Manager that is a powerful tool for implementing object-oriented database management systems. This layered approach can provide transparency and reusability.

The remainder of this paper is organized as follows. Section 2 describes an overview of EGG GIS. In section 3 and 4, we present the User Interface layer and the Data Manager layer, respectively. In section 5, we describe the EXODUS Storage Manager layer that is developed by Wisconsin university. Finally, we conclude the paper and discuss future work in section 6.

2. Overview of EGG

Geographic information systems are used for applications whose users usually are not experts in operating computers, such as atmospheric phenomena, police action, military, cartography, demography, town/urban planning, navigation, and natural resource management. EGG supports a graphic-based interactive user interface that the beginner can use easily to get a useful information from the stored geographic data.

The scenario of obtaining result objects from EGG is as follows:

- Select a source object by clicking the left mouse button on a map. If there are more than one object on that point, then the user can select a source object in the browsing list.
- Select classes whose objects can be results.
- Perform an aspatial query or press the spatial operator button such as closest, contains, range, etc.
- Perform a view or zoom operation as needed.
- Save current intermediate results for the future usage.
- Click the right mouse button for a direct search at any time.

In this paper, we design and implement EGG using a layered approach. Figure 1 shows the overall architecture of EGG.

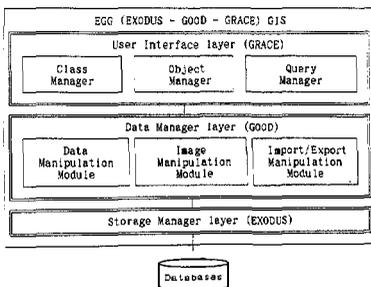


Figure 1: Architecture of EGG

There are several types of application programming interfaces (APIs). The most general approach in relational database management systems is the use of a query language, because there is a standard query language (i.e., SQL) for relational database management systems. Unlike relational database management systems, there is no standard for the geographic database management system. Furthermore, the query language approach requires additional time to preprocess the query language and to map the GIS data to/from host variables. That is, the query language approach is not suitable for an API of the geographic database management systems. Therefore, in this paper, we implement a callable function library as an API of our geographic database management system.

In this paper, we use object-oriented concepts such as domain, class, and object: the domain is a logical unit that constitutes a GIS, the class is defined with common attributes, and the object is geographic data that belong to a class.

3. GRACE (GRAPhic User InterfaCE for a GIS)

In this paper, we design and implement GIS with an emphasis on easy-of-use. That is, we build a graphic-based interface on X-Windows for user friendly interfaces.

3.1 Class Manager

The Class Manager enables a system administrator to define classes that are used in the geographic information system and to define attributes to be included within each class. Figure 2 shows a user interface of the Class Manager.

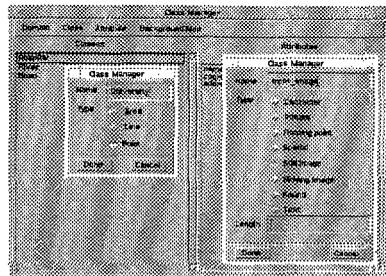


Figure 2: User interface of the Class Manager

The Class Manager provides the following menus:

- DOMAIN menu is used to select a particular domain.
- CLASS menu is used to create, update,

and delete classes. Point, line, and area types are supported as a class type.

- **ATTRIBUTE** menu is used to create, update, and delete attributes. Character, integer, float, spatial, still image, moving image, sound, and text types are supported as an attribute type. Character type needs an attribute size to hold fixed-length data.
- **BACKGROUND MAP** menu is used to specify a file name for importing a bitmapped background map. The PCX graphic file format which is used widely on PC based softwares is supported as a background map format. In this paper, a background map is treated as a class which has only one object. Therefore, users can display a background map as well as classes.

3.2 Object Manager

The Object Manager enables a system administrator to manage objects in the geographic information system. Figure 3 shows a user interface of the Object Manager.

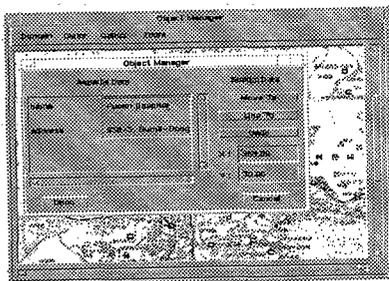


Figure 3: User interface of the Object Manager

The Object Manager supports the following menus:

- **DOMAIN** menu is used to select a domain to manage. Since EGG supports a distributed environment, selected domains can reside at other hosts.
- **CLASS** menu is used to select a class whose objects are inserted.
- **OBJECT** menu is used to insert, update, and delete objects. An administrator can insert spatial data with a mouse on the graphic screen and aspatial data with a form-filling method.
- **ZOOM** menu is used to zoom in and out a map.

3.3 Query Manager

The Query Manager enables end users to get a useful information from the geographic information system. Figure 4 shows a user

interface of the Query Manager.

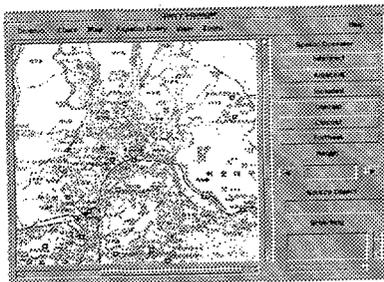


Figure 4: User interface of the Query Manager

The Query Manager provides the following menus:

- **DOMAIN** menu is used to select a domain from which users can obtain an information. Since EGG supports a distributed environment, selected domains can reside at other hosts.
- **CLASS** menu is used to select classes to be displayed.
- **MAP** menu is used to save a current map and load intermediate results as a map. 'Overlay' for retrieving, 'union' for OR subquery, and 'intersection' for AND subquery are supported as load methods.
- **ASPATIAL** menu is used to perform an aspatial query using GUI. In order to perform an aspatial query, a user selects a class and an attribute, specifies a value by using a form-filling method, and then presses the comparison operator button such as equal, bigger than, not, etc.
- **VIEW** menu is used to specify a color of objects corresponding to the attribute value for the efficient visual analysis.
- **ZOOM** menu is used to zoom in and out a map.

In addition to the above menus, 'direct searching' that enables a user to get attribute values of an object by clicking the right button on a map and 'browsing' for selecting an object from the intermediate results are supported by the Query Manager.

4. GOOD (Geo-Object Oriented Data Manager)

The Data Manager layer plays a role of performing operations defined by the User Interface layer efficiently. It consists of three modules. The data manipulation module manages a spatial index for spatial data and connects the User Interface layer with the Storage Manager layer. The image manipulation module presents spatial data on the graphic screen. The import/export manipulation module imports a bitmapped background map into EGG

GIS and exports the intermediate results to the UNIX file system. Since the Data Manager layer is a callable function library, it can be used easily by the Use Interface layer.

4.1 Data manipulation module

The data manipulation module performs functions that connect the User Interface layer with the Storage Manager layer and convert data formats of the User Interface layer into actual stored data formats of the Storage Manager layer [15].

The data manipulation module consists of initialization functions, domain manipulation functions, class manipulation functions, field manipulation functions, and object manipulation functions. Initialization functions are called not only at the beginning of the Data Manager layer but also at the end of it. Domain manipulation functions are used to select a desired domain from the domains connected by a network. Class manipulation functions are used for the management of classes defined in each domain. Field manipulation functions are used to manage fields contained within each class. Object manipulation functions are used for the management of geographic objects that belong to each class.

4.2 Image manipulation module

Unlike other types of computer applications, GIS manages spatial data. Therefore, the Data Manager layer has the image manipulation module to support a display of spatial data on the graphic screen. If the data manipulation module and the image manipulation module are used to develop a user interface for other GIS applications, it can reduce its developing time.

The image manipulation module is divided into zooming functions and object presentation functions. In this paper, all of geographic data are stored and managed as vectors. Therefore, zooming functions are used to control the overall size. Object presentation functions display geographic objects that are stored in a geographic database on the graphic screen.

4.3 Import/export manipulation module

Since there are some needs to interface with the UNIX file system, we implement the import/export manipulation module. The import/export manipulation module is used to import a bitmapped background map into EGG GIS and exports the intermediate results to the UNIX file system.

The import/export manipulation module is divided into importing function and export function. The file format that is used to import a bitmapped background map is the PCX file format as mentioned earlier. The file format that is used to export the intermediate results is

the PostScript file format which is independent of the existing operating systems and used widely for printing.

5. EXODUS Storage Manager

The Storage Manager layer efficiently stores and manages large amounts of geographic data which need large space and consist of variable-length records. Therefore, widely used relational database management systems are not suitable for GIS because programmers have to make more efforts to supply facilities which are not supported in relational database management systems. Therefore, in this paper, we use the EXODUS Storage Manager which is a tool for implementing object-oriented database management systems. Figure 5 shows the structure of the Storage Manager layer.

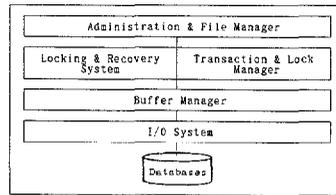


Figure 5: Architecture of the EXODUS Storage Manager

The EXODUS Storage Manager is a flexible storage manager that provides concurrent and recoverable access to stored objects of arbitrary size. That is, it is a collection of access methods that provide associative access to files of stored objects and support for versioning. For access methods, EXODUS provides a library of type-independent index structures including B+ trees and linear hashing. In addition, it provides libraries of database system components for access methods, version management, and simple operations. Since EXODUS supports the client/server architecture, upper layers of GIS at one host can access geographic databases on other hosts in a distributed environment.

6. Conclusion

Clearly, the desire for developing GIS in various applications is great. Since, however, the users of GIS may not be an expert in using computers, traditional user interfaces (for example, SQL in relational database management systems) are not suitable for GIS. Furthermore, graphic facilities must be supported for presenting the spatial data. Therefore, we design and implement the geographic information system with a graphic user interface.

Using a database management system is essential for developing a geographic

information system to store and retrieve geographic data. Therefore, we design and implement the geographic database management system (GDBMS) using the EXODUS Storage Manager which is a powerful tool to develop an object-oriented database management system. On top of this GDBMS, we design and implement a graphic-based interactive user interface which enable users to manage and access geographic data in GIS easily. In this paper, we use object-oriented concepts to model geographic data efficiently.

The architecture of GIS is layered into three parts, and in each of these layers we selected a method of using development tools. The User Interface layer was implemented by using OSF/Motif based on the X-Windows with an emphasis on easy-of-use. In the Storage Manager layer which stores geographic data in a database, we used the EXODUS Storage Manager that is a tool for implementing an object-oriented database management system. In the Data Manager layer, we adopt a callable function library as an application programming interface (API) in order to transform a lot of variable-length geographic data between the Storage Manager layer and the User Interface layer.

An 3-dimensional representation of geographic objects and an interface with GPS (Global Positioning System) for the dynamic management of moving objects can be considered as the future work.

References

- [1] Andrew, U.F., 1991, "Beyond Query Languages for Geographic Databases : Data Cubes and Maps," Proc. of the Geographic Database Management Systems Workshop, Italy, pp.5-17.
- [2] Berman, R., and Stonebraker, M., 1977, "GEO-QUEL: A System for the Manipulation and Display of Geographic Data," Computer Graphics, Vol.11, No.2, pp. 186-191.
- [3] Carey, M., DeWitt, D., and Shekita, E., 1989, "Storage Management for Objects in EXODUS," in Object-Oriented Concepts, Databases, and Applications, Kim, W. and Lochovsky, F., eds., Addison-Wesley.
- [4] David, B., Raynal, L., and Schorter, G., 1993, "Geo2 : Why Objects in a Geographical DBMS ?" Proc. of the 3rd Int. Symp. Spatial Database Systems, Springer Verlag, pp. 264-276.
- [5] Egenhofer, M.J., 1994, "Spatial SQL : A Query and Presentation Language," IEEE Trans. on Knowledge and Data Engineering, Vol.6, No.1, pp. 86-95.
- [6] Faloutsos, C., and Kamel, I., 1993, "High Performance R-trees," IEEE Data Engineering, Vol.16, No.3, pp. 28-33.
- [7] Gunther, O., and Rieker, W.F., 1993, "The Design of GODOT : An Object-oriented Geographic Information System," IEEE Data Engineering, Vol.16, No.3, pp. 4-9.
- [8] Haas, L., and Cody, W., 1991, "Exploiting Extensible DBMS in Integrated Geographic Information Systems," in Proc. of the 2nd Symposium Spatial Database Systems, Springer Verlag, pp. 423-449.
- [9] Jonathan, F.R., and Michael, S.B., 1991, "Implementation of a User Environment for a Spatial DBMS," Proc. of the Geographic Database Management Systems Workshop, Italy, pp.18-33.
- [10] Medeiros, C.B., and Pires, F., 1994, "Databases for GIS," ACM SIGMOD, Vol.23, No.1, pp. 107-115.
- [11] Morehouse, S., 1992, "The ARC/INFO Geographic Information System," Computers and Geosciences : An International Journal, Vol. 18, No.4, pp. 435-443.
- [12] Oh, B.W., and Han, K.J., 1993, "Design and Implementation of the Geographic Information System Using an Object-Oriented Database Management System," Proc. of KISS Autumn Conf., Seoul, Korea, pp. 129-132.
- [13] Oh, B.W., and Han, K.J., 1994, "GRACE : A Graphic User Interface for Geographic Information Systems," Proc. of KISS SIGDB Winter Database Symp., Pusan, Korea, pp. 58-64.
- [14] Oh, B.W., Han, K.J., and Kim, S.Y., 1994, "The Design and Implementation of a Geographic Information System for the Efficient Management of Work-path," Proc. of 4th Korea Telecom Int. Symp., Taejon, Korea, pp. 178-185.
- [15] Oh, B.W., Lee, W.Y., and Han, K.J., 1994, "GOOD : Data Manager for Geographic Information Systems," Database Review, Vol.10, No.2, KISS SIGDB, pp. 3-22.
- [16] Ooi, B.C., 1988, Efficient Query Processing in a Geographic Information System, Ph. D. Thesis, Monash University.
- [17] Scholl, M., and Voisard, A., 1991, "Object-oriented Database Systems for Geographic Applications : An Experiment with O2," Proc. of Geographic Database Management Systems, Capri, pp. 103-137.
- [18] Tanaka, M. and Ichikawa, T., 1988, "A Visual User Interface for Map Information Retrieval Based on Semantic Significance," IEEE Trans. on Soft. Eng., Vol.14, No.5, pp. 666-670.
- [19] Williamson, R., and Stucky, J., 1991, "An Object-Oriented Geographical Information System," in Object-Oriented Databases with Applications to CASE, Networks, and VLSI CAD, Prentice-Hall, pp. 296-312.
- [20] Yasuaki, N., and Shigeru, A., 1993, "Interactive Graphics and Spatial Data Management for GIS Using the Hierarchical Data Structure," Proc. of the Far East Workshop on GIS, Singapore, pp. 106-120.