GeO2: Object-Oriented Contribution
for a Geographical DBMS?

Benoit DAVID, Laurent RAYNAL, Guylaine SCHORTER
IGN / COGIT & CNRS / Cassini, 2 Avenue Pasteur, BP 68
F-94160 Saint-Mandé, France
E-mail: {david,raynal,schorter}@cogit.ign.fr

Véronique MANSART
FLEXIMAGE / BDG, 43 rue de la Brèche-aux-Loups
F-75012 Paris, France
E-mail: vero@-inf.enst.fr

Abstract. This paper proposes a semantic geographical data model. Localization of geographical entities is defined using an abstract data type. The geographical entities' semantic is expressed through high level concepts in an entity-relationship model extended by both inheritance and propagation mechanisms.

The system's implementation is described. It takes advantage of the object-oriented aspect of the O2 DBMS, and in particular of the inheritance mechanism. Three different data structures allow to implement geographical entities' localization. They correspond to the topology description levels described in modern exchange formats. A principle of independence between data structures and operations is defined. This principle allows avoiding the re-questioning of the definition of a process if the data structure must be modified.

Lastly the described system has been effectively developed and is used with sets of a large volume of real data produced by the French Institut Géographique National (IGN).

1 Introduction

To date, most of the Geographical Information Systems (GIS) available on the market are built above a Relational Data Base Management System (RDBMS). The RDBMS manages the entire set of classical data types - which we call facts. At the same time, a proprietary file management system takes care of geometrical and/or topological data. A complex pointer mechanism translates links between facts and geometry. As a first consequence, this system is a single-user one since the file manager is single-user. Furthermore, there is an impedance mismatch between the two manager levels and the two query languages are not homogeneous. Arc/Info from ESRI Corp. [21] and MGE from Intergraph Corp. are such systems.

A more recent approach consists in building a GIS by extending a RDBMS. As a first step the RDBMS is extended to manage geographical objects by adding spatial data types and associated spatial functions and predicates. This architecture is more suitable for geographical data management. It eliminates previous disadvantages such as single-user, impedance mismatch and non-homogeneous query languages. System 9
from Computer Vision [9], Smallworld [4], GEO++ [25] and GeoSabrina [20] are such systems. Nevertheless, the RDBMSs are still poor regarding several aspects: concepts are simple; the data type system is weak; data have to be normalized in first normal form while hierarchical structures are needed; semantic links are lost and need to be rebuilt through integrity constraints; and the data access system is very expensive because of joins.

A third very recent approach consists in building the GIS around an Object-Oriented DBMS (OODBMS). All Object-Oriented (OO) programming benefits are available such as object identity, complex objects, property inheritance and methods. However, even if the OODBMS architecture seems to be more appropriate, it is still necessary to extend the system for managing geographical data. Specifically, geometrical data types (i.e., point, line and area) must be added to its basic structure and associated functions (union, distance, rectangle, ...), and predicates (cross, adjacency, overlapping, ...) must be added to the query language.

The work presented in this paper belongs to this approach. The objective was to add the necessary functions to make a GIS using a commercially available OODBMS. We chose O2 from O2Technology [2] as a basis of our work. Similar works have been described in [17, 24] for example. But our work has unique aspects.

First of all, this system, we name: GeO2, is truly operational and has been used with sets consisting of large volumes of real geographical data containing several tens of thousands of lines and several hundreds of thousands of points (40 Megabytes). Performance was therefore a main objective of our work.

On the other hand, since different data sets may be structured differently, a GIS must be capable of handling the different data structures. A GIS should not mandate its own data structure because a change of structure is never neutral, either more information is necessary, or information is lost. The described system offers different data structures and induces independence between data structures and the operations. This independence allows the operations not to be modified if data structures must. The operations need to be programmed only once, even if they must be computed with different data structures.

This paper is organized as follows. Section 2 reviews the numerous studies about the benefits of OO in geographical DBMS. Section 3 introduces our data model and section 4 describes the implementation of the data model. Section 5 concludes this paper.

2 Background

The term "object", as applied to geographical data management, has generated numerous papers these past years. Topics addressed in these papers, associated with this fashionable word, can be classified in two different categories which correspond to two approaches toward the object-oriented phenomenon.

In the first category, the OO phenomenon is considered as a technique, used for software engineering. In [19, 13, 4, 22], the benefit of OO languages for geographical DBMS software development is discussed because OO concepts have originated from