Object-Oriented Data Modelling for Spatial Databases

Kritika Mishra¹, Pravin S Metkewar²

¹Symbiosis Institute of Computer Studies & Research (SICSR), A constituent of Symbiosis International University, Pune - 411016, Maharashtra, INDIA
²Associate Professor, Symbiosis Institute of Computer Studies & Research (SICSR), A constituent of Symbiosis International University, Pune - 411016, Maharashtra, INDIA

Abstract—Data modelling is a critical stage of database design. Recent research has focused upon object-oriented data modes, which appear more appropriate for certain applications than either the traditional relational model or the entity-relationship approach. The object-oriented approach has proved to be especially fruitful in application areas, such as the design of geographical information systems which have a richly structured knowledge domain and are associated with multimedia databases. This article discusses the key concept in object-oriented modelling and demonstrates the applicability of an object-oriented design methodology to the design of geographical information systems. In order to show more clearly how this methodology may be applied, the paper considers the specific object-oriented data model.

Keywords: IFO, Spatial database, Semantic, Fragment.

I. INTRODUCTION

The choice of an appropriate representation for the structure of a problem is perhaps the most important component of its solution. For database design, the means of representation is provided by the data model. A data model provides a tool for specifying the structural and behavioral properties of a database and ideally should provide a language which allows the user and database designer to express their requirements in ways that they find appropriate, while being capable of transformation to structures suitable for implementation in a database management system. Data modelling is among the first stages of database design. The purpose of data modelling is to bring about the design of a database which performs efficiently; contains correct information (and which makes the entry of incorrect data as difficult as possible); whose logical structure is natural enough to be understood by users; and is as easy as possible to maintain and extend. Of course, different problems require different means of representation and a large number of data models is described in the database literature. Some are close to implementation structures.

II. Object-Oriented Data Modelling

In describing the above abstraction constructs, we have gradually moved towards an object-oriented view. In the basic E-R model, the entities are conceived as having attributes, occurrences of which are drawn from atomic domains. That is, the underlying domains are of basic and indecomposable types such as integer, real and string. As we bring in the abstractions above, we add a further dimension to the structure of the underlying domains, which no longer need be atomic. In object-oriented data modelling, all conceptual entities are modelled as objects. An abstraction representing a collection of objects with properties in common is called an object type. Objects of the same type share common functions. The objects associated with an object type are called occurrences. Integer and string are object types, as is a complex assembly such as a city. Indecomposable object types are called primitive. Decomposable objects are called composite or complex objects. A composite object, therefore, is an object with a hierarchy of component objects. We have seen how complex types may be formed from primitive types using generalization, specialization, aggregation and grouping. These are the primary Object-Oriented Data Modelling for Spatial Databases object-type
operations in object-oriented data modelling. Other operations have been introduced and can be found in the literature. Object-oriented data models support the description of both the structural and the behavioral properties of a database. Structural properties concern the static organizational nature of the database. Behavioral properties are dynamic and concern the nature of possible allowable changes to the information in the database. This paper concentrates on the structural description. The object-oriented approach to data modelling has proved to be especially fruitful in application areas which are not of the standard corporate database type. Complex molecular and engineering part-assembly databases are examples of systems which have been successfully modelled using these techniques. What such applications have in common is a richly-structured semantic domain, often with a hierarchical emphasis, and associated with multimedia database (e.g., text, numeric, graphical, audio).

III. OBJECT TYPES
IFO is truly object-oriented in that all its component types may be composite. Atomic types are of three kinds; printable, abstract and free. A printable type corresponds to objects which may be represented directly as input and output. Examples of printable types are integer, string, real and pixel. An abstract type corresponds to physical or conceptual objects which are not printable. Person is an example of an abstract type. Free types serve as links in generalization and specialization relationships. Representations of examples of atomic types indicate string and integer types respectively. Non-atomic types are constructed from atomic types using aggregation and grouping. For diagrammatic clarity, it is sometimes convenient to treat complex types as atomic.

IV. FUNCTIONAL RELATIONSHIPS BETWEEN OBJECTS
So far the ways in which complex objects may be constructed from atoms have been described. We now discuss how types may be related. IFO provides a formalism for representing functional relationships between types. The means by which functional relationships are represented is the fragment. Informally, a fragment is a part of the IFO model, containing types and functions (but no generalization or specialization relationships), subject to certain constraints. We illustrate with an example, shown in Figure 1. This fragment shows functional relationships SITUATION and CITIZENS between object types CITY, REGION and PEOPLE. The structure of the knowledge being modelled here is that cities are situated in regions and are occupied by people, each of whom may have for an address a location which is an aggregation of a house number and street name. The function CITIZENS has the dependent function ADDRESS. Intuitively, this models the case where a person may live in more than one city and so have different addresses in different cities. The E-R model allows the possibility of many-valued relationships between types and so appears to be more general than a functional model. However, the grouping operator can be used to provide the facility of representing many-valued functions. For example, the relationship shown in Figure 6.1, where a person may live in several cities and a city comprises many people, is represented functionally in IFO as shown in Figure 6.5, where the image of a city under the function CITIZENS is a set of persons, since it is an object of type PEOPLE, which is an association of PERSON.
V. CONCLUSIONS

This paper has traced the development of data modelling from the relational model to a contemporary object-oriented method and suggested, with examples, some of the applications of object-oriented modelling to geographical information systems. It is argued that such methodologies offer clear advantages over traditional methods such as E-R modelling. In particular, object-oriented modelling allows database designers to incorporate more readily the complexities of spatial data.

REFERENCES