An Object Repository Model for the Storage of Spatial Metadata

Sophie Cockcroft[1]
Department of Information Science
University of Otago

June 1995

Abstract

The design of spatial information systems has traditionally been carried out independently of mainstream database developments. It is contended that the adoption of mainstream database design techniques is important to progress in the spatial information systems development field. An accepted approach to the development of information systems is through an integrated development environment with a design repository at its core. This paper proposes a skeleton model for the design of a repository to store spatial metadata. An object oriented modelling approach is adopted in preference to an entity relationship approach because of its ability to model functional and dynamic aspects of the repository.

[1] Address correspondence to: Ms S.K.S. Cockcroft, Lecturer, Department of Information Science, University of Otago, P.O. Box 56, Dunedin, New Zealand. Fax: +64 3 479 8311 Email: scockcroft@commerce.otago.ac.nz

1 Introduction

Progress in the spatial information system (SIS) industry has traditionally occurred independently of advances in mainstream non-spatial database technology. [1]. Further, the focus of database design in spatial information systems has frequently been the acquisition of data, often with no specific use in mind [2]. Tools such as database management systems (DBMS), fourth generation languages (4GLs) and computer-aided software (or system) engineering (CASE) are commonplace in the context of developing non-spatial information systems. DBMS remove the need to specify data structures in application programs. DBMS also provide mechanisms for the management of relationships between data items, as well as the implementation of integrity constraints. Fourth generation languages, with tools for standardised user interface design, non-procedural programming and code-generation based on data dictionary specifications, support the use of prototyping approaches to system development. CASE tools provide modelling methods for the specification of system design. Design repositories populated by CASE tools can then be used as the basis for database schema and code generation. The automation of such modelling methods within CASE environments allows the derivation of database schema definition language from graphical representations of data requirements (ie the derivation of logical schema from conceptual schema).

Tools for spatial information systems development are, by comparison, relatively crude. Results of research into spatial data structures and research into mapping spatial data into records in relational databases have found their way into commercial implementations of geographical information systems. There is no commercial database system today that *directly* supports spatial data management, in particular, data definition and query facilities for spatial data [3]. Currently available spatial data management systems are typically one of two types: either they are hybrid in nature where spatial (ie 'map') data are stored separately from descriptive data, or they are based on some form of extended relational data model where a series of tables are pre-defined to store spatial data [2]. In either case, it is usually necessary for application programmers to manage the consistency and currency of linkages between spatial and non-spatial data. The majority of existing GIS are implemented using relational databases, and typically adopt a hybrid architecture [4]. Since this approach does not maintain an entire map base within a single DBMS, geometric data are not subject to the same rigorous management as is applied to non-spatial data. There is a good case for developing unified extended relational systems with spatial and non-spatial data under the same architecture [4]. However, Healey [5] identified a number of problems inhibiting the development of such systems, chiefly in the area of performance overheads. Recent moves in the commercial sector indicate that spatial data management is becoming a topic of interest to the major database vendors [6]. A related issue from the point of view of spatial querying is the emerging SQL3 standard. This moves to unify object and relational data. As relational technology advances it is adopting many of the constructs associated with the object oriented paradigm.

In summary, it is desirable that many of the features widely available for non-spatial data management are incorporated in spatial data management systems. A more rigorous approach to spatial database design is undoubtedly required. This requirement provides the motivation for the current work. The specific focus of this research is the design of a repository for the storage of spatial metadata within an integrated spatial data management environment, and the design decisions associated with the development of such a repository.

2 A design repository for spatial metadata

Three closely related terms to describe the place where metadata is stored are the data dictionary, the metadatabase, and the repository. There are numerous definitions of these three, and the meanings of the terms often overlap. It would appear that these three terms differ in broadness. The narrowest term is the metadatabase. This is commonly used in the field of GIS and has been described as a catalogue of existing data sets [7]. Such a catalogue has also been developed in [7]. The role of a repository has also been described as ‘to manage datasets’ [8]. The implication of this definition is that the datasets have already been collected and the database must ‘fit around them’. Data dictionary is a somewhat broader term, describing computer software that records, stores and processes descriptions of an organisation’s data resources [9]. This term is often used in the database context. A data dictionary is described as holding detailed information about objects within a database including the tables and views, and the columns within those tables and views. The broadest definition is the repository. A repository stores data about the objects that are within a database and also supports process and project management [10]. According to Cheng [11] the purposes of a repository include applications development in a CASE environment, enterprise information resource management and information processing and management at a global level within the enterprise. Its contents comprise metadata that model both data resources and knowledge.

The repository is essentially a specialised database that stores ‘data about data’ or metadata. Here, a repository design using Rumbaugh’s Object Modelling Technique (OMT) [12] is proposed. A rationale for the adoption of this methodology was given in [13]. The major motivations for the adoption of OMT were that it allows for abstract data typing and the incorporation of behavioural aspects into models.

3 Introduction to OMT

The Object Modelling Technique decomposes an application into three types of model: the object model which describes the structure of objects in the system; the dynamic model which describes those aspects of the system concerned with time and the sequencing of operations; and the functional model which describes those aspects of a system concerned with transformations of values.

One aspect of a system is described by each model and the models are linked to one another by references. The object model describes the data structure on which the dynamic and functional models operate. The operations in the object model correspond to events in the dynamic model and functions in the functional model. The dynamic model describes the control structure of objects. It shows decisions that depend upon object values and invoke functions. The functional model describes functions invoked by operations in the object model and actions in the dynamic model. Functions operate on data values specified by the object model. The functional model also shows constraints on object values. The relationship between the three models is shown in Figure 1 below.

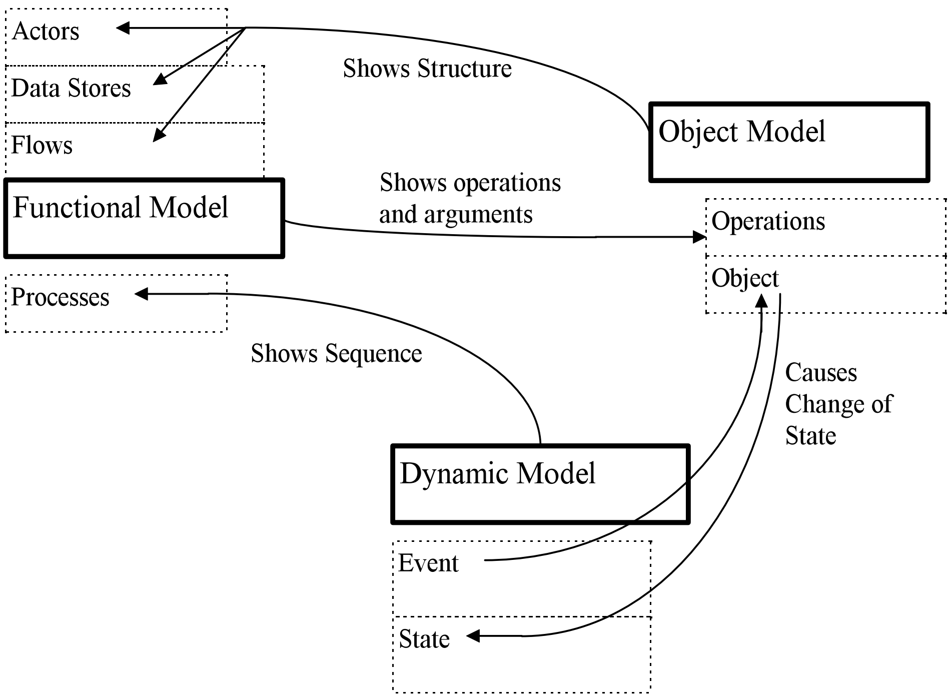

Figure 1: Relationship between object, dynamic and functional models

4 OMT and repository design

The remaining sections concern the incorporation of desirable features into repository design through the use of the three models of OMT. A coarse overview of the problems to be addressed by each model is given.

It is desirable that the repository should be active not passive, that is it should be possible to impose run-time controls on the data being entered into the system through the use of the repository. This necessitates the modelling of dynamic behaviour at some stage in the repository design process. Object oriented concepts such as encapsulation and inheritance can be used to associate behavioural properties with entity sets. This makes object modelling an appealing option since they allow for the inclusion of such concepts at the modelling stage.

The roles of the three OMT models, when applied to repository design, are as follows: the object model would be concerned with structural aspects such as the range of topological relationships that can be specified; the dynamic model would be concerned with the development of the repository over time, for example the management of updates, deletions and version control; and the functional model would be concerned with schema and report generation from the repository.

4.1 The object model

The draft object model for the repository design is given in Figure 2. With regard to this model the following points should be noted.

Relationships are known as associations in the OMT terminology. Associations may be binary, ternary or higher order. In practice relationships higher than ternary are very rare. However, some ternary relationships cannot be broken down into binary associations without losing information. In order to allow for n-ary relationships in the repository model, ROLES are introduced. An ENTITY plays a ROLE in a relationship. Each relationship aggregates two or more roles (a binary relationship would be composed of 2 roles, a ternary relationship of 3 roles, etc.).

Spatial and non-spatial relationships are disjoint subclasses of the superclass RELATIONSHIP, whereas entity sets with spatial characteristics and entity sets with non-spatial characteristics are non-disjoint subclasses of the superclass ENTITY. This reflects the fact that a given entity may exhibit both spatial and non-spatial characteristics.

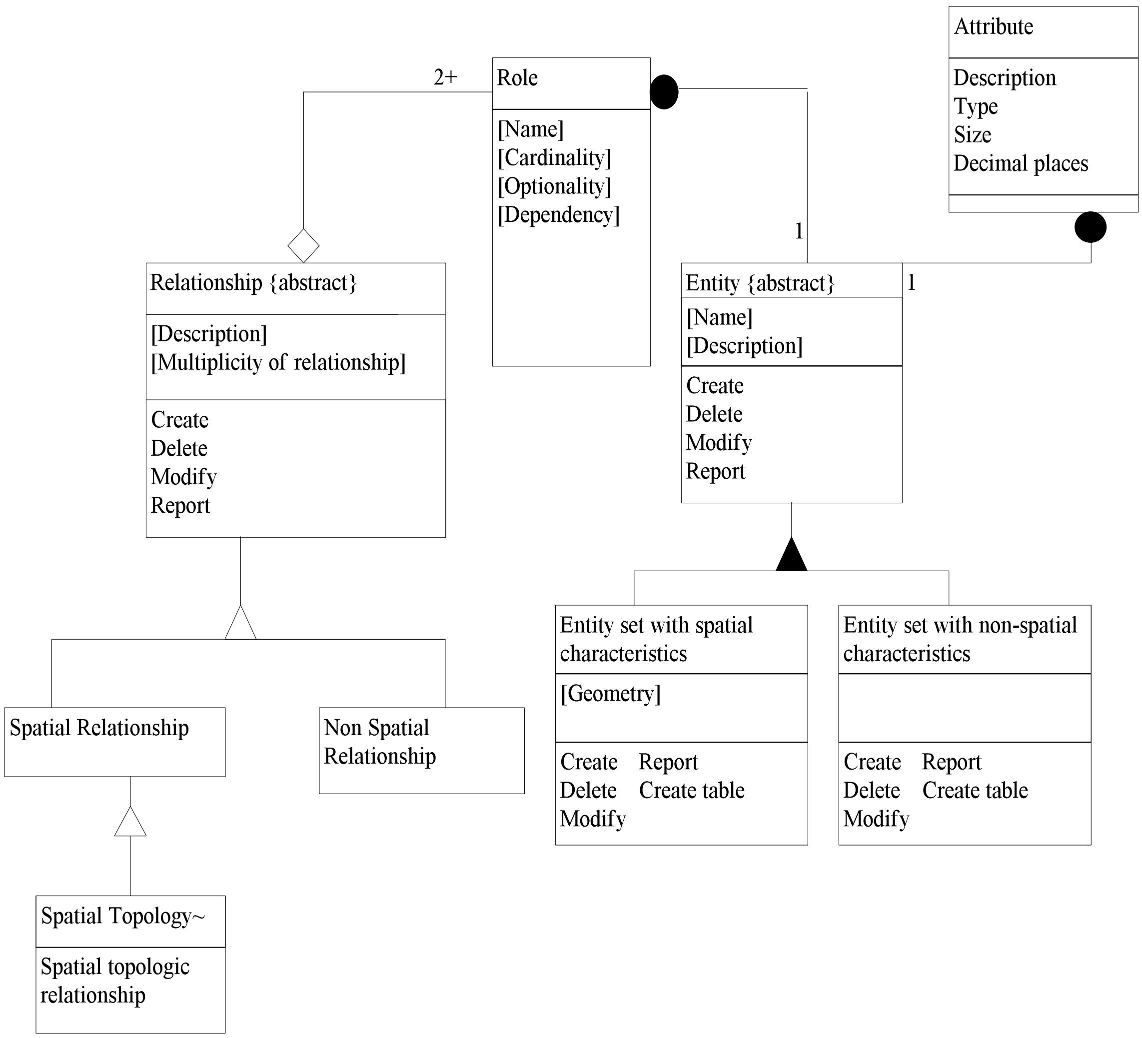

Figure 2: Repository object model

Spatial relationships are further classified into spatial topological relationships. These correspond to those spatial relationship types specified by the set algebra of Egenhofer and Franzosa [14] and given in Table 1 below. The entities are the base geometric types from which user oriented entities will be specified.

Entity Type	*Spatial Relationship Type*	
line	contain	spatial grouping
point	spatial aggregation	boundary
region	cover	
arc	overlap	
	disjoint	
	point adjacent	
	line adjacent	

Table 1: Correspondence between entity and spatial relationship types

The inclusion of a topology of spatial relationships, and the implications for generating logical level schema based on this topology is the key spatial aspect of the repository.

4.2 The functional model

In OMT, the functional model is described in terms of data flow diagrams. The major functional aspects of the repository are a general report-generating facility and the generation of logical schema. These are illustrated in Figure 3. More specific treatment of this functional component will centre on connections between conceptual modelling methods and existing logical models for the storage of spatial data - analogous to the derivation of logical database schema from conceptual schema. A specific example of this approach would be to investigate connections between the SEER model [2], the only known conceptual level spatial data model, and the "cartographic modelling methodology" [15], which is essentially a logical level spatial data model. It is thought that a study of a data definition language such as this may provide a useful first step toward understanding linkages between conceptual and logical models of spatial data, and hence form the basis for the derivation of algorithms for logical schema generation from the spatial design metadata held in the repository.

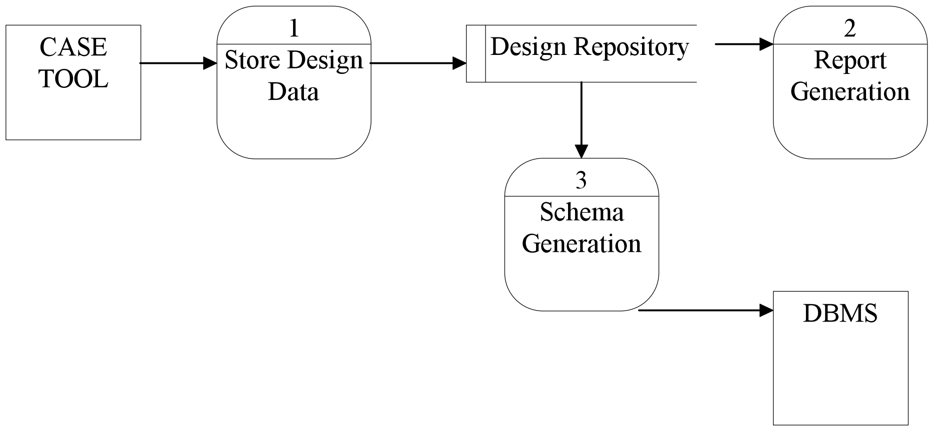

Figure 3: Functional model

4.3 The dynamic model

During the design phase, the major dynamic aspect of the repository would be handling updates to the design. An additional dynamic aspect would be the performance of consistency checks on data models created with the CASE tool used to populate the repository. However, it is desirable that the repository should be active after implementation of the spatial information system and, in this context, the dynamic aspects would be more far reaching. The repository would have to handle updates in the database managed by the DBMS. The dynamic model of OMT uses state diagrams to express changes to objects and their relationships over time. One state diagram exists for each class with important dynamic behaviour. In a relational database context one approach to recording changes to an entity with geographical attributes over time is to employ a tuple based time-stamping approach for the storage of such historical data [16]. A new tuple or row is added to the relevant relation as each change is made. This approach supersedes approaches that necessitated the addition of a new version of the relation each time an update was made. The incorporation of this approach into the dynamic modelling of entity and relationship objects over time is under consideration as a means of update handling in the repository.

5 Conclusion

In conclusion, it is felt that consideration of the object modelling technique has contributed to the clarification of spatial metadata repository design problems. A structural model for repository design has been suggested. In addition design issues relating to dynamic and functional aspects have been discussed. In an integrated software development environment there are clear advantages to uniformly applying one paradigm to design each part of the system. The object oriented paradigm is suggested as the appropriate approach to repository design and implementation, not only for reasons of uniformity, but also expressibility. The next stage of research will be the expansion of the functional model to include algorithm design for the generation of data definition language statements. The dynamic model will also be enhanced to explicitly support the management of changes in object classes over time. A full breakdown of metadata necessary to describe conceptual level data models is also required. The storage of data about spatial relationships and the derivation of algorithms to generate logical data models for the storage thereof is expected to be the key originality of this work.

References

1. Milne, P., Milton, S. and Smith, J.L. "Geographical object oriented databases - a case study." International Journal of Geographical Information Systems. 7(1): 39-55, 1993

2. Firns, P. G. An Extended Entity Relationship Model Applicable to the design of Spatially Referenced Databases. PhD Thesis, University of Otago 1994.

3. Kim, W., J. Garza and A. Keskin. "Spatial data management in database systems: research directions." Advances in Spatial Databases. Third International Symposium, SSD '93 Proceedings. Abel ed. 1993 Springer-Verlag Berlin, Germany.

4. Worboys, M. F. “Object Oriented approaches to geo-referenced information” International Journal of Geographical Information Systems (8) 4, 385-399, 1994

5. Healey R.G. "Database Management Systems”, in Maguire et al “Geographical Information Systems, Longman 1991. " 1: 251-67.

6. Oracle 7 MultiDimension - Oracle Press release April 1995 - http:\\www.oracle.com

7. Radke, John and Yvonne Everett. "The Klamath Meta Data Dictionary." Annual Meeting of the American Society of Geographers, San Fransisco, CA April. 1994

8. Shoshani A, "Experience with creating a metadatabase and a metadata browsing tool" IEEE workshop on metadata for scientific and technical data management May 1994 Washington DC

9. Henderson, M. “The Importance of data administration in information management” Information Management Review 1987 2(4) 41-47

10. Brown, C “The role of a repository in the adoption of object-oriented techniques and databases” CASE Trends July 1993

11. Cheng, H., M. Bouziane, L. Rattner and L. Yee. "Information Resources Managament in heterogeneous distributed environments: A metadatabase approach." IEEE transactions Software Eng. 17(6): 1991.

12. Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy and W. Lorensen. "Object-oriented modelling and design." 1991 Prentice Hall. New Jersey.

13. Cockcroft, S.K.S. “Towards the Specification of a Spatial Metadata Repository Structure” Proceedings of the 1995 New Zealand Conference on Geographical Information Systems and Spatial Information Research Massey University, April 26th - 28th 1995. 97-111

14. Egenhofer, M. J., and Franzosa, R. D. "Point set topological Spatial relations" International Journal of Geographical Information Systems. 5(2): 161-174, 1991.

15. Tomlin, C. D. "Geographic Information Systems and Cartographic modelling ." 1990 Englewood Cliffs (New Jersey: Prentice Hall).

16. Raafat, H., Z. Yang and D. Gauthier. "Relational spatial topologies for historical geographical information." International Journal of Geographical Information Systems. 8(2): 163-173, 1994