7042_book.fm Page 119 Friday, May 12, 2006 2:41 PM

International Journal of Geographical Information Systems,
1990, Vol. 4, No. 4, 369–383.

6 Object-Oriented Data Modelling for Spatial Databases

Michael F. Worboys, Hilary M. Hearnshaw, and David J. Maguire

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 IFO. Standard cartographic primitives are represented using IFO, which are then used in the modelling of some standard administrative units in the United Kingdom. The paper concludes by discussing current research issues and directions in this area.

1 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 behavioural 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,

7042_book.fm Page 120 Friday, May 12, 2006 2:41 PM

for example the relational model (Codd 1970). Others as yet have no directly corresponding implementation. This is the case for those which support a wide variety of modelling constructs as well as a high level of abstraction. Such models allow representations which are closer to the original problem as framed by the user. An example is the IFO (Is-a relationships, Functional relationships, complex Objects) model discussed later.

In this paper, emphasis is placed upon the so-called object-oriented data models, which are at the problem-oriented end of the scale. Object-oriented approaches originated in programming languages such as Simula and Smalltalk. The application of object-oriented ideas to databases was spurred on by the apparent limitations of traditional relational technology when applied to some of the newer applications. Typical examples are the applications of databases in computer-aided design (CAD), office information systems (OIS), software engineering and geographical information systems (GIS). A common difficulty in all of these application areas is the gulf between the richness of the knowledge structures in the application domains and the relative simplicity of the data model in which these structures can be expressed and manipulated. Object-oriented models have the facilities to express more readily the knowledge structure of the original application.

There is no clear definition of, or even general agreement in the computing community on what precisely is, an object-oriented data model. The area is still very new and individual ideas have not yet been synthesized into a general view. However, it is generally recognized (Peckham and Maryanski 1988) that there is a clear ascending chain from the relational model through the earlier semantic data models to object-oriented models and it is in this context that this paper considers object-oriented modelling methods. The major abstraction constructs are discussed and exemplified. One particular object-oriented formalism, IFO, developed by Abiteboul and Hull (1984) is considered, followed by a description of the application of the IFO formalism to two examples in the GIS area. These examples show how IFO can represent precisely the basic spatial elements (point, line and polygon) and, on a larger scale, represent and model relationships between administrative and postal area units in the United Kingdom. Such modelling is a key application for any GIS (see, for example, Egenhofer and Frank 1989).

To avoid confusion, it is necessary to differentiate object-oriented data modelling from object-oriented database management systems (OODBMSs). An OODBMS is a system upon which the database is implemented. It is possible (but not optimal) to model using an object-oriented methodology and implement in, for example, a relational DBMS. Of course, it is most desirable to use an OODBMS which can naturally implement all the constructs of the data model. However, owing to the newness of the technology, OODBMSs are only now emerging as viable systems. A recent description of some of the most innovative of such systems (e.g., Iris, ORION, OZ +, and GemStone) is given in Kim and Lochovsky (1989). Such systems will have an important impact on GIS technology. For example, most OODBMSs support version control, where the system can generate multiple different versions of an object, maybe corresponding to different time-slices. This would be a natural implementation of spatially referenced data (e.g., census data) where spatial boundaries

7042_book.fm Page 121 Friday, May 12, 2006 2:41 PM

can change. After this brief mention of the emergent OODBMS technology, the remainder of this paper will concentrate on the object-oriented data models.

2 Semantic data models

The relational model (Codd 1970) provides the database designer with a modelling tool which is independent of the details of physical implementation. However, the relational model is limited with respect to semantic content (i.e., expressive power) and there are many design problems which are not naturally expressible in terms of relations. Spatial systems are a case where the limitations become clear. To illustrate this point, consider the relational model of a polygon as originally given by van Roessel (1987), based upon the definitions proposed by the National Committee for Digital Cartographic Data Standards (Moellering 1986) and discussed in greater detail later in this paper:

POLYGON (Polygon, ID, Ring ID, Ring Seq)
RING (Ring ID, Chain ID, Chain Seq)
CHAIN1 (Chain ID, Point ID, Point Seq)
CHAIN2 (Chain ID, Start Node, End Node, Left Pol, Right Pol)
NODE (Node ID, Point ID)
POINT (Point ID, X Coord, Y Coord)

This model of a polygon as a set of relations, though complete, is low-level and some way from one which represents a user's normal view of such an object. Semantic data models aim to provide more facilities for the representation of the users' view of systems than the relational model, as well as to de-couple these representations from the physical implementation of the databases. Fundamental work in this area was undertaken by Chen (1976), who proposed a semantic data model and a diagrammatic technique known as the entity-relationship (E-R) model and diagram respectively. The concepts underlying the E-R model are described in some detail in the next section, since these concepts are required for an understanding of many later semantic data models.

3 Entity-relationship modelling

The entity-relationship model utilizes the concepts of entity, attribute and relationship. A distinction is made between a type and an occurrence of a type. An entity is an item about which the database is to record information. Such an item should be uniquely identifiable. For example, a particular point could be uniquely identified by its coordinates or a census tract by its census code. An entity type is an abstraction representing a class of entities of the same kind. For example, POINT and CITY are entity types. Occurrences of those types are particular points and cities, e.g., a point with coordinates (3,4) and a city named Oxford.

An attribute is an element of data associated with an entity. A city has a population, thus the entity type CITY has attribute type POPULATION. (In this

7042_book.fm Page 122 Friday, May 12, 2006 2:41 PM

FIGURE 6.1 Entities and relationship.

paper, types will usually be printed in upper-case.) A particular city has a particular population. Such a population is an example of an attributed occurrence. To avoid a cumbersome presentation, we will omit the terms 'type' and 'occurrence' when no ambiguity is involved. The attribute(s) which identify an occurrence of an entity uniquely are termed identifiers or keys.

A relationship is an association between entities. For example, LIVES_IN is relationship between entities PERSON and CITY. Again, we may distinguish between types and occurrences of relationships. Relationships may have attributes, for example, the relationship LIVES_IN might have the attribute DURATION which gives the length of time that a person has lived in a city.

Chen (1976) proposed a diagrammatic means of representing this model. The diagrammatic form of the above example is shown in Figure 6.1. Rectangles depict entities and rhombi depict relationships. M and N indicate that the relationship is many-to-many, that is each person may live in more than one city and each city may have more than one person living in it.

Many systems may be modelled using entities, attributes and relationships, including systems with a dominating spatial component. Calkins and Marble (1987) apply the method to the design of a cartographic database. They describe the strengths of the method as being flexibility, control of database integrity and generality (i.e., not linked with particular implementations). An important feature of E-R modelling is the natural and well-understood method of the transformation from the E-R model to the rational model (Whittington 1988).

4 Extensions to the entity-relationship model

The entity-relationship approach is at present recognized as the prime tool for data modelling (see, for example, Whittington 1988). However, experience has shown that for many systems the initial set of modelling constructs (entity, attribute and relationship) is inadequate. For example, view integration (the process by which several local views are merged into a single integrated model of the database) is recognized by many workers (for example, Calkins and Marble 1987) as of great importance for GIS design. View integration is greatly facilitated by the introduction of abstraction concepts additional to the original E-R model. In the mid-1970s, Smith and Smith (1977) proposed the introduction of two abstraction constructs, generalization and aggregation, into the data modelling tool-kit. These constructs are provided by almost all contemporary semantic data models. We proceed to describe them in detail, (using some of the IFO notation which is explained in detail later in the paper), distinguishing between generalization and specialization, considering a further construct, association or grouping, and then discussing their relevance to spatial database design. A similar approach will be found in Egenhofer and Frank (1989).

7042_book.fm Page 123 Friday, May 12, 2006 2:41 PM

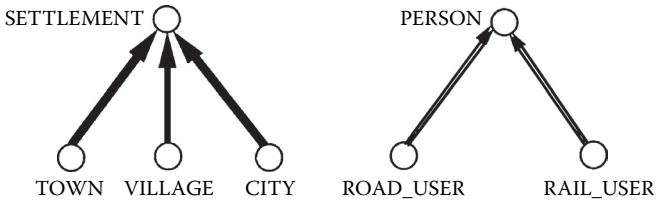

FIGURE 6.2 SETTLEMENT is a generalization of TOWN, VILLAGE and CITY. ROAD_ and RAIL_USER are specialization of PERSON.

4.1 Generalization

Generalization is the construct which enables groups of entities of similar types to be considered as a single higher-order type. For example, entities of types VILLAGE, TOWN, and CITY may be merged and considered as entities of the single type SETTLEMENT. SETTLEMENT is said to be a generalization of VILLAGE, TOWN, and CITY. The diagrammatic representation of this situation is shown in Figure 6.2. Formally, the generic higher-order type is the set-theoretic union of objects in the lower-order types. An object may be thought of, at this intermediate stage between classical and object-oriented data modelling, as an entity along with its attributes. This definition will be extended and made more formal later in the paper.

4.2 Specialization

Specialization is the construct which enables the modeller to define possible roles for members of a given type. For example, entities of type PERSON might be considered occurrences of type ROAD_USER or RAIL_USER, depending upon the context in which we see them. The diagrammatic representation of this situation is shown in Figure 6.2. Formally, the specialized type is made up of a subset of occurrences of the higher-order type.

It should be noted that, although generalization and specialization are in a sense inverse to one another, there are distinctions. A generic type inherits its structure from its lower-order types (and possibly adds some of its own). In the case of specialization, the specific types inherit structure from the higher-order type (and possibly add some of their own). In the case of both generalization and specialization, we say that the lower-order type is a subtype of the higher-order type.

4.3 Aggregation

Aggregation is the construct which enables types to be amalgamated into a higher-order type, the attributes of whose objects are a combination of the attributes of the objects of the constituent types. Formally, the objects which are occurrences of the aggregate type are tuples, the components of which are the objects of the constituent types. In short, aggregation corresponds to the mathematical operation of cartesian product. An example is the type POINT, which is the aggregate of type POINT_ID with two integer types named X_COORD and Y_COORD, thus representing a point

7042_book.fm Page 124 Friday, May 12, 2006 2:41 PM

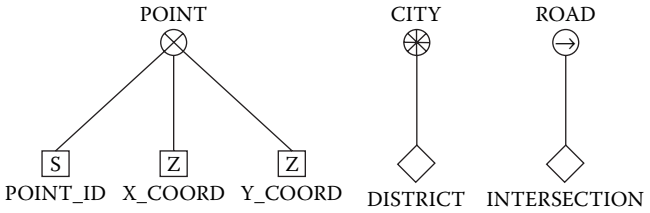

FIGURE 6.3 POINT is an aggregation of identifier and coordinates. CITY is an association of DISTRICT. ROAD is an ordered association of INTERSECTION.

as having two spatial coordinates. This relationship is represented diagrammatically in Figure 6.3.

4.4 Association

Association or grouping is the construct which enables a set of objects of the same type to form an object of higher level type. It is often stipulated that the sets are finite. The corresponding set-theoretic construct is the power-set operator. An example is the view of a city as, amongst other things, a collection of districts. CITY is an association of DISTRICT. This is shown diagrammatically in Figure 6.3.

4.5 Ordered association

Sometimes, it is important to take into account the ordering of a collection of objects. For example, the ordering of intersections making up a road may be critical. Ordered sets are called lists and we allow a higher-order type which is a collection of lists of the lower-order type. In our example we say that ROAD is an ordered association of INTERSECTION. This is shown diagrammatically in Figure 6.3.

5 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

7042_book.fm Page 125 Friday, May 12, 2006 2:41 PM

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 behavioural properties of a database. Structural properties concern the static organizational nature of the database. Behavioural 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). Since a GIS also shows these characteristics, it seems then that a GIS is an ideal application for object-oriented modelling.

In order to show more clearly how this methodology may be applied, we will concentrate on the specific recent data model IFO (Abiteboul and Hull 1987) which contains the above object-oriented constructs. It is concerned almost wholly with structural properties of a database.

6 IFO

The IFO model was introduced by Abiteboul and Hull (1984). A more condensed account of the model is given by Abiteboul and Hull (1987). IFO incorporates all the constructs so far introduced in this paper with the exception of 'relationship'.

6.1 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 (shown in IFO by a square) 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 (shown in IFO by a diamond) corresponds to physical or conceptual objects which are not printable. PERSON is an example of an abstract type. Free types (shown in IFO by circles) serve as links in generalization and specialization relationships. Representations of examples of atomic types are given in Figure 6.4. S and Z indicate STRING and INTEGER types respectively. Non-atomic types are constructed from atomic types using aggregation and grouping as already discussed. For diagrammatic clarity, it is sometimes convenient to treat complex types as atomic. For example, Figure 6.3 shows POINT is an aggregate of atomic types but later diagrams treat POINT as abstract atomic.

FIGURE 6.4 Atomic types.

7042_book.fm Page 126 Friday, May 12, 2006 2:41 PM

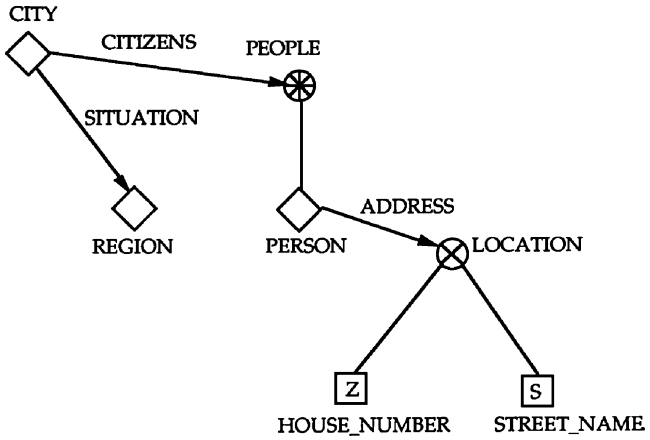

FIGURE 6.5 The CITY fragment if IFO.

6.2 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 6.5. 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. Formally, a fragment F is a rooted directed tree of types. The root of a fragment is called a primary vertex. Full details of fragments may be found in Abiteboul and Hull (1984).

6.3 Schemas

In IFO, fragments form the building blocks of schemas. A schema is the largest IFO unit and is a forest of fragments, possibly connected together at their primary vertices

7042_book.fm Page 127 Friday, May 12, 2006 2:41 PM

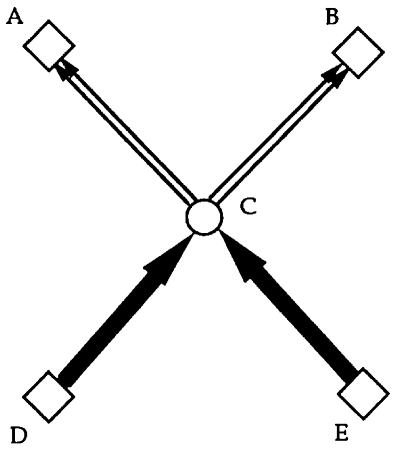

FIGURE 6.6 An inconsistent structuring of objects.

by generalization and specialization edges. Thus the schema allows the representation of all the components of the IFO model.

IFO places some constraints on the way that schemata may be constructed. These concern the directions in which objects are structured from other objects. For example, aggregated and grouped objects are constructed from their constituent elements. The non-commutativity of this structuring relationship leads to the distinction made between generalization and specialization. A generalized type is structured from its sub-types. A specialized type is structured from its super-types. The schema must have its object sources and sinks arranged in a consistent fashion. For example, the configuration shown in Figure 6.6 is not permissible as it results in inconsistent structuring of objects. Objects of type C result from specializing objects of types A and B and generalizing objects of types D and E. There is no guarantee that this can be done consistently and that clashes will not result. A further point here is that, even neglecting D and E, C is the result of specializing from two possibly quite distinct types A and B. Again, there is no guarantee that this can be done consistently and we would require that types A and B 'arise' from a common type. If all the arrows in Figure 6.6 were reversed, we would again have an impermissible schema. In this case, C is a source for objects of types A, B, D and E, but is not itself defined in terms of any other type. It could be considered a 'black hole' of the system. Abiteboul and Hull define the permissible configurations by stating five rules which they must satisfy. They also state a theorem showing that any schema satisfying their rules leads to a consistent structuring of objects at each vertex of the schema. The reader is referred to Abiteboul and Hull (1984) for details.

7 Fundamental spatial objects if IFO

The first application of IFO that we present is its use to represent the three fundamental spatial object types; point, line, and polygon. These representations are based upon the definitions proposed by the National Committee for Digital Cartographic

7042_book.fm Page 128 Friday, May 12, 2006 2:41 PM

Data Standards (Moellering 1986), which are summarized in van Roessel (1987) as follows

A point is a zero-dimensional spatial object with coordinates and a unique identifier within the map.
A line is a sequence of ordered points, where the beginning of the line may have a special start node and the end a special end node.
A chain is a line which is a part of one or more polygons and therefore also has a left and right polygon identifier in addition to the start and end node.
A node is a junction or endpoint of one or more lines or chains.
A ring consists of one or more chains.
A polygon consists of one outer and zero or more inner rings.

Section 2 shows how it is possible to represent these spatial elements directly using the relational model. An IFO representation of POINT is given in Figure 6.3 and that for NODE, LINE and POLYGON in Figures 6.7, 6.8 and 6.9, respectively. Figure 6.7 shows a node as a special kind of point, with its own node identifier as well as its identifier as a point. In Figure 6.8, a line is modelled as an ordered association of points, with identifier and begin and end nodes. A polygon, in Figure 6.9, is an ordered association of rings, which in turn are ordered associations of chains. Polygons, rings and chains have identifiers. A chain is a special type of line with corresponding left and right polygons.

The aim is a presentation which accords with users' own views of an object and decouples the representation from the implementation.

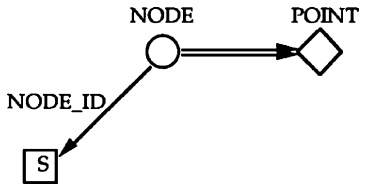

FIGURE 6.7 NODE modelled in IFO.

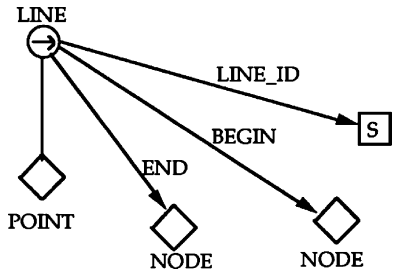

FIGURE 6.8 LINE modelled in IFO.

7042_book.fm Page 129 Friday, May 12, 2006 2:41 PM

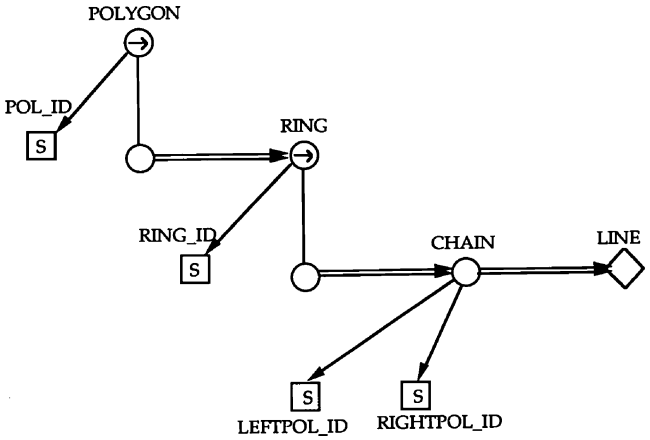

FIGURE 6.9 POLYGON modelled in IFO.

8 Spatial units for Leicestershire in IFO

The second and more substantial application is the modelling of spatial units and their relationships using our object-oriented formalism. Leicestershire is a county located in the Midlands, England. Its central city, Leicester, is the administrative and industrial heart of the county, with half a dozen county towns around the perimeter (Figure 6.10). At the Midlands Regional Research Laboratory (MRRL), in setting up a geographical database for Leicestershire which contains a wide range of data sets covering the county, one of the major problems is the diversity of spatial units on which the data are reported (Hearnshaw et al. 1989). The relationships between the spatial units bearing data and the lack of those relationships are of significance in the design of the database. This section gives examples of IFO models of the post-code-based and census/administrative units.

8.1 The post-code system

The United Kingdom is divided into 120 post-code areas, of which Leicester is one, designated LE. The post-code area, LE, is divided into 17 district post-codes (LEI to LE17). These in turn are divided into 86 sector post-codes (LE1 7, LE16 6, etc.). Sectors are divided into the smallest of the post-code areal units: the unit post-code (UPC) e.g., LE1 7RH, which is a unique identifier for all the points of delivery (addresses) on one postman's 'walk'. The post-code address file (PAF) provides the UPC for all addresses in the county. It also provides a coded form of the address: a 4-digit PREMCODE, which consists of the first four characters of information of an address, e.g., 3 Main Street has PREMCODE 3MAI. There are provisions in the PREMCODE for removing ambiguities, and so this, together with the UPC, can uniquely identify every address (Post Office 1985). It can be seen (Figure 6.11) that these units nest neatly into each other in a clearly defined, and clearly identifiable,

7042_book.fm Page 130 Friday, May 12, 2006 2:41 PM

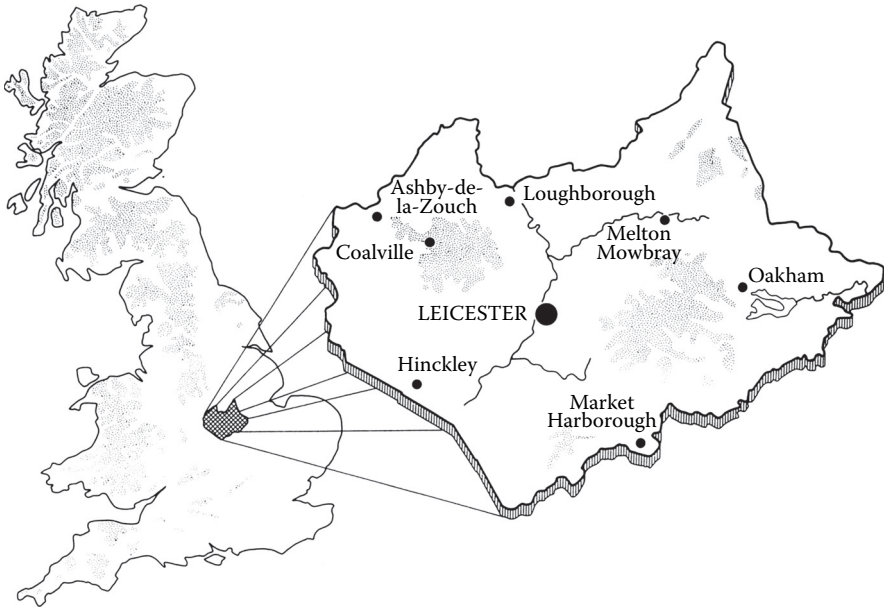

FIGURE 6.10 Map of Great Britain showing Leicestershire (Strachan 1985). Shaded areas show land above 200 m.

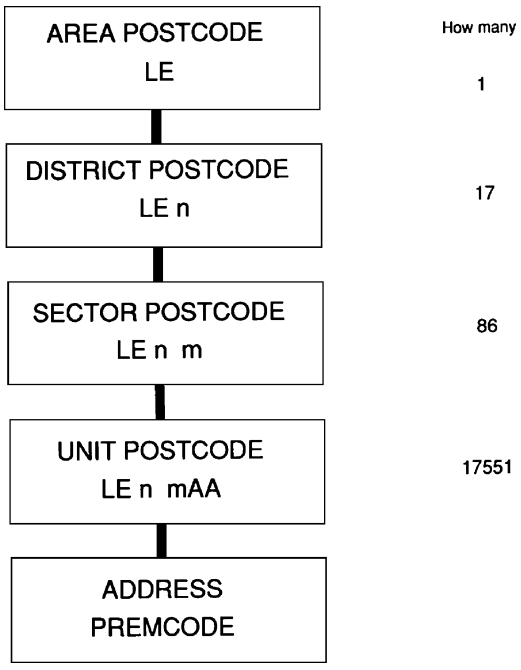

FIGURE 6.11 The post-code units for Leicestershire.

7042_book.fm Page 131 Friday, May 12, 2006 2:41 PM

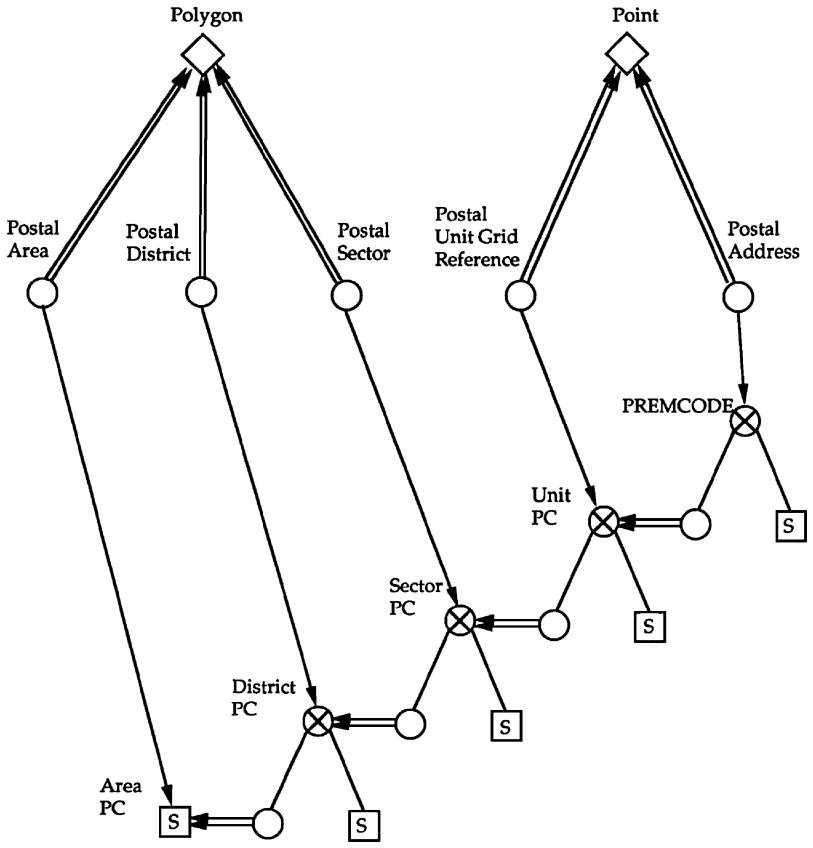

FIGURE 6.12 Relationship between postal units in IFO.

hierarchy. The Central Postcode Directory (CPD) provides the Ordnance Survey Grid reference for the SW corner of a 100m grid square for the first address in each UPC.

Figure 6.12 displays the postal units in IFO. POLYGON and POINT types, defined earlier, are taken as given abstract atomic types. It can be seen that each postal unit type (for, example, POSTAL DISTRICT) is a specialization of an abstract spatial type (for example, POLYGON), and is also the domain of a function whose co-domain is a composite string type which constitutes an identifier (for example, DISTRICT POSTCODE). This representation makes a clear and useful distinction between the spatial and non-spatial aspects of the postal units.

8.2 Census of population and local administrative units

The most recent, decennial Census of Population was taken in 1981. The basic spatial unit on which census data are collected and published in England and Wales is the Enumeration District (ED). The size of an ED varies from about 500 households, in densely populated urban areas, to about 150, in rural areas. The 1981 EDs were

7042_book.fm Page 132 Friday, May 12, 2006 2:41 PM

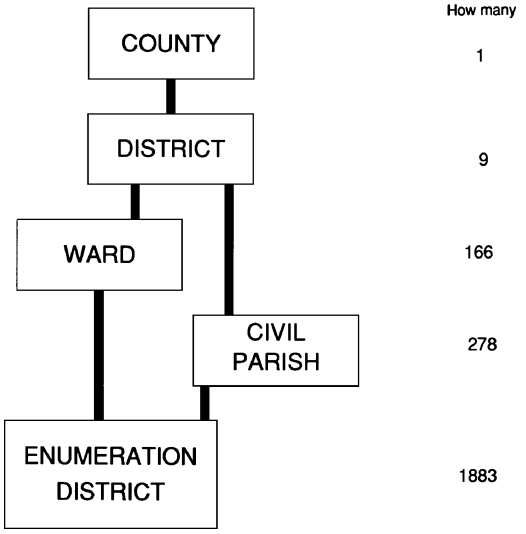

FIGURE 6.13 Some census and administrative units for Leicestershire.

designed to partition district electoral wards, that is, a collection of non-overlapping EDs exactly cover each ward. Wards also partition the districts. The units nest neatly into each other in a well-defined, and clearly identifiable, hierarchy, as shown in Figure 6.13. In the case of Leicestershire, those parts which used to be rural districts are divided into civil parishes. The 1981 EDs were also designed to partition civil parishes.

The Small Area Statistics (SAS) files of the 1981 Census of Population include a 12-digit grid reference, for the weighted centre population (centroid) for each ED. Thus the ED data can, in theory, be related to other spatial units, such as post-codes, via grid references (Gattrell 1988).

Each ED is uniquely identifiable by a six character code. The leftmost two characters define the district, the middle two characters define the ward, and the rightmost two define the ED. For example, ED code ABCD04 defines the fourth ED in ward CD of district AB. Each county also has a 2-digit code: that for Leicestershire is 32.

The administrative units corresponding to the 1981 census are represented in IFO in Figure 6.14. COUNTY, DISTRICT, WARD and PARISH are specializations of POLYGON. ENUMERATION DISTRICT has a CENTROID which is of type POINT. All the units have NAMEs, and there exist functions from all unit types, except PARISH, to identifiers which are nested as shown in the diagram. PARISHes may be associated with sets of enumeration district identifiers, (i.e., identifiers for those EDs which are contained in each parish).

An example of the relationships between postal and census units is given in Figure 6.15 where the functional relationship between postal units and wards provided by the Central Postcode Directory is given. With each postal unit is associated a unique ward.

7042_book.fm Page 133 Friday, May 12, 2006 2:41 PM

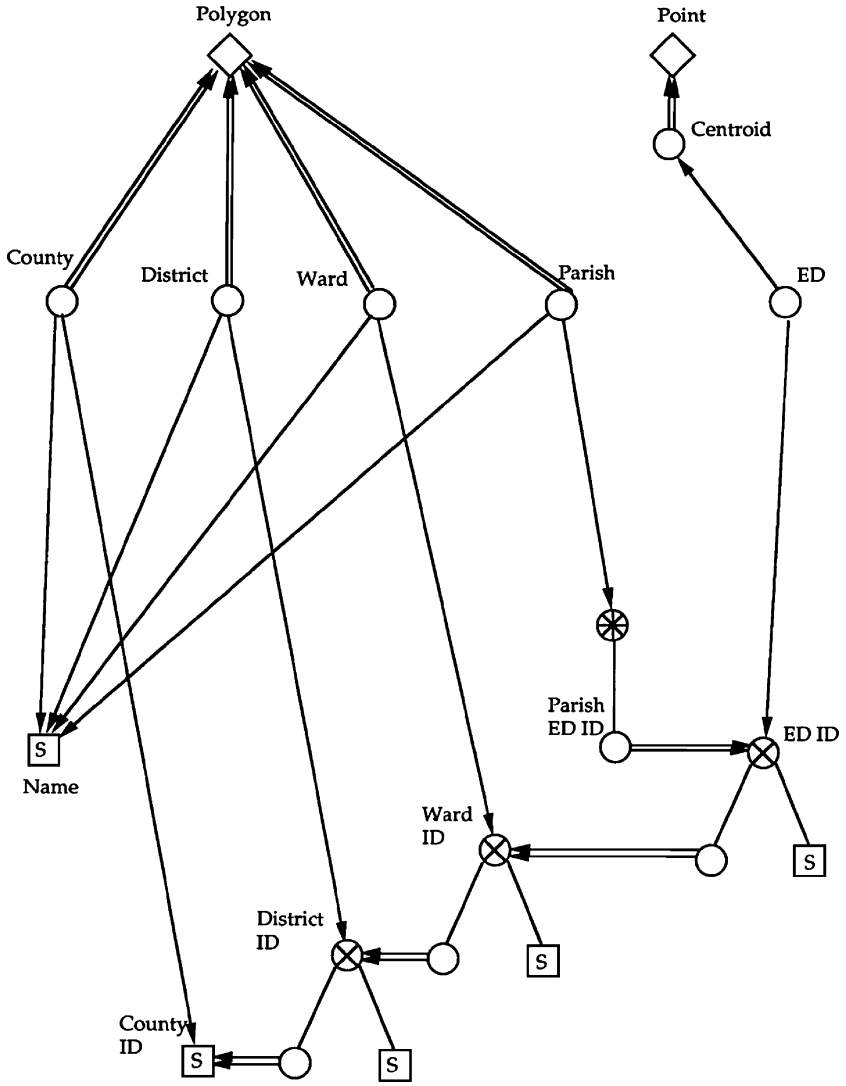

FIGURE 6.14 Relationship between 1981 census units in IFO.

FIGURE 6.15 Functional relationship between post-codes and wards.

7042_book.fm Page 134 Friday, May 12, 2006 2:41 PM

Relationships between different spatial units are considerably complicated by boundary redefinitions, for example, corresponding to different census years. These problems are handled in object-oriented systems by means of version control, where facilities exist for the definition of different versions of an object.

9 Implementation and research issues

The discussion up to this point has focused upon data modelling in an object-oriented setting. When the modelling is complete there follows the implementation stage. The system designer must then decide upon the DBMS which will implement the application. At present, the relational systems are by far the most popular for most applications, using standard commercial software such as IBM's DB2, ORACLE, INGRES and others. Under development, and shortly due on the market, are object-oriented systems. These provide a direct link with the modelling tools discussed here and, once proved efficient, will be ideal implementation tools.

If a relational DBMS is chosen for implementation, the object-oriented data model must be transformed into the relational model. It has been shown (Blaha et al. 1988) that many of the constructs discussed in this paper can be mapped to the relational model, although losing some of their natural meaning for the user and some efficiency in the process. However, a unified treatment of the relationship between IFO and the relational model has not yet been given.

At present, the user of a spatial database that is implemented on a general-purpose DBMS has available only the general purpose query language provided by the system. The industry standard query language SQL (Structured Query Language) is satisfactory for interrogating databases of the standard corporate type, but is less than ideal for databases with a complex object structure. It would be better to have a language which matched the data model in some way, with facilities for handling spatial and object-oriented constructs. This research area promises to be fruitful (Frank 1988). There are two main approaches to the construction of a spatial query language. Either one starts from fundamentals or one builds on top of an existing language such as SQL. At present, the latter is the favoured approach. However, it is well recognized within the database community (see, for example, Whittington 1988) that SQL itself has shortcomings. The authors are engaged in the construction of a spatial query language using the former approach.

A major concern of computer science at present is the problem of correctness of systems. Computers are now used to control many systems where safety is the highest priority. The correct functioning of software developed for such safety-critical systems should be guaranteed. The methodologies being developed to deliver such guarantees are based upon the formal specification of systems, including database systems. Research needs to be undertaken which leads to the formal specification of GIS. The IFO model, since it is itself formally defined, is an excellent vehicle for such work. Already, Abiteboul and Hull (1984) have described work which shows the effects of updates on the integrity of object-oriented systems. Research is in progress at the MRRL which considers these questions in the special light of spatial databases.

7042_book.fm Page 135 Friday, May 12, 2006 2:41 PM

10 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.

Acknowledgment

Acknowledgment is made of the help provided by colleagues Alan Strachan and Bill Hickin of the MRRL, and Graham Winter of the Research and Information Group, Leicestershire County Council Department of Planning and Transportation.

References

Abiteboul, S., and Hull, R., 1984, IFO: A Formal Semantic Database Model, TR-84-304, Computer Science Department, University of Southern California.

Abiteboul, S., and Hull, R., 1987, IFO: A formal semantic database model. *Association for Computing Machinery Transactions on Database Systems*, **12,** 525.

Blaha, R. B., Premerlani, W. J., and Rumbaugh, J. E., 1988, Relational database design using an object-oriented methodology. *Communications of the Association for Computing Machinery*, **31,** 414.

Calkins, H. W., and Marble, D. F., 1987, The transition to automated production cartography: design of the master cartographic database. *American Cartographer*, **14**, 105.

Chen, P. P.-S., 1976, The entity-relationship model–toward a unified view of data. *Association for Computing Machinery Transactions on Database Systems*, **1**, 3.

Codd, E. F., 1970, A relational model of data for large shared data banks. *Communications of the Association for Computing Machinery*, **13,** 377.

Egenhofer, M. J., and Frank, A. U., 1989, Object-oriented modeling in GIS: inheritance and propagation. *Proceedings of Auto-Carto 9 held in Baltimore, Maryland, in April 1989*, edited by E. Anderson (Falls Church: American Congress on Surveying and Mapping American Society of Photogrammetry and Remote Sensing), pp. 588–598.

Frank, A. U., 1988, Requirements for a database management system for a GIS *Photogrammetric Engineering and Remote Sensing*, **54,** 1557.

Gattrell, A. C., 1989, On the spatial representation and accuracy of address-based data in the U.K., *International Journal of Geographical Information Systems*, **3**, 335.

Hearnshaw, H. M., Maguire, D. J., and Worboys, M. F., 1989, An introduction to area-based spatial units: a case study of Leicestershire. Midlands Regional Research Laboratory Research Report No. 1, University of Leicester.

Kim, W., and Lochovsky, F. H. (editors), 1989, *Object-Oriented Concepts, Databases, and Applications* (Berkshire: Addison-Wesley).

Moellering, H., 1986, A review and definition of 0-, 1- and 2-dimensional objects for digital cartography. *Proceedings of the 2nd International symposium on Spatial Data Handling* (Ohio: International Geographical Union Commission on Geographical Data Sensing and Processing).

7042_book.fm Page 136 Friday, May 12, 2006 2:41 PM

Peckham, J., and Maryanski, F., 1988, Semantic data models. *Association for Computing Machinery Computer Surveys*, **20**, 153.

Post Office, 1985, *The Postcode Address File Digest* (London: Post Office).

Smith, J. M., and Smith, D. C. P., 1977, Database abstractions: aggregation and generalization. *Association for Computing Machinery Transactions on Database Systems*, **2**, 105.

Strachan, A. J., 1985, Atlas of Leicestershire, County and Employment Research Unit, University of Leicester.

van Roessel, J. W., 1987, Design of a spatial data structure using the relational normal form. *International Journal of Geographical Information Systems*, **1,** 33.

Whittington, R. P., 1988, *Database Systems Engineering* (Oxford: Oxford University Press).