Modelling Ecological Data

Peter J Barclay and Jessie B Kennedy
Napier University,
Craiglockhart Campus,
219 Colinton Road,
Edinburgh EH14 1DJ,
Scotland.
e-mail:
pete@uk.ac.napier.cs and jessie@uk.ac.napier.cs

Abstract

Ecological surveys generate large quantities of data; database technology has not yet reached its full potential in this area. Here we investigate approaches to modelling ecological data, considering the requirements for a successful model. An object oriented conceptual model is presented, and applied to the results of an actual survey. A data management system based on this object oriented approach is briefly described.

KEYWORDS: data modelling, ecological data, object orientation.

0 Introduction

Many kinds of information gathering, such as ecological surveys, result in the collection of large volumes of data. Statistical techniques are used to manipulate this data and perform analyses upon it [Gau82], [Pie84]; meanwhile, the relevance of these collections of numbers to the real world phenomena which they describe may be somewhat obscured. This article investigates whether object oriented data modelling techniques may provide means of representing such data in which the data's meaning is preserved in a more intuitive way. It is not suggested that such an approach would replace established techniques, but rather that it might form a useful supplement to them.

The closest type of data to that considered here is that used by geographic information systems (GISs) (see [Abd91] for a review of state-of-the-art in GISs). These systems typically treat spatially-referenced data which also has aspatial properties. Articles such as [WHM90] use semantic data models to define such basic building blocks as points, lines and polygons.

The chief difference with ecological data is that it is less geometrical in nature, and the structure of the data reflects the survey methodology used to collect it. In [RGHH91], it is proposed that GIS data should be considered from 3 complementary aspects called the *geometric, overlay* and *feature* views. Using this terminology, the surveyed pixels discussed in section 5.1 comprise the geometric view, and the overlay and feature views are simply constrained collections and aggregations of these pixels respectively. The spatial-referencing in this data is implicit.

Difficulties inherent in handling ecological survey data are explored in [Ken85]. Here, we first review such data in general in an abstract way, and consider what features must be present in a successful model; we might view this section as a requirements specification for the model. We then go on to develop a general model, which is expressed in an object oriented framework. Finally, we apply this approach to modelling some data from a real survey, and briefly describe a system implementing this application model.

1 Survey Data

A survey collects information about the real world. We might picture the existence of a survey site, in which we are interested but about which we have no information. During the survey, we go out and collect information;

the more extensive the survey, the more information we will gather, though we will never know everything there is to know about the site.

We may wish to combine the results of a survey with other surveys over the same area to increase our understanding; we may wish to organise other follow-up surveys based on considerations of the initial results. Since field work is time consuming, we wish to get the best possible use from the data we have gathered.

1.1 Survey Methodology

Many different methods of collecting survey data exist. Usually only some subsection of the site of interest can be investigated, with the hope that the results gathered will be of general applicability to the entire site. Observations may be for the presence or absence of a given feature, or may involve quantifying or categorising some feature. Hence discretisation (an essential part of formal approaches to modelling GIS data such as in [Wor92]) is automatically performed *before* the data is modelled.

In general, each survey will comprise of a set of observations. These observations pertain both to an actual area of ground, and to some feature of interest. We might observe that a sampling area contains 23 spiders, or that it is fenny. Our method of handling the data must allow us to represent areas of the Earth's surface, and facts about them which are of interest to us.

In this article we concentrate on the collection, modelling and presentation of survey data in general, paying little attention to what the features of this data are, since in different surveys these might vary widely.

1.2 Observations

In a survey, observations are conducted to amass information about the survey site. We mean these observations to be the validation of statements we make about the site. If for example we observe fen in an area A, then we refer to the fen as a *feature* of interest in A, to the statement 'fen was observed at A' as the *observation predicate,* and the statement 'A is fenny' as the *feature predicate.*

1.2.1 Limiting Assumptions

It is, of course, possible that an observation might be in error, which might be an issue in conducting real life surveys. However, let us assume for a simple life that all our observations are accurate, so that given an observation, we may without error affirm the corresponding observation predicate. Let us further assume that we may safely affirm the feature predicate corresponding to a given observation predicate over the same area. This is the *truth of observation* assumption (1).

It is also assumed that any observations are not invalidated by the passage of time; this is a less tenable assumption, since we may well be interested in the time evolution of the survey site. Thus two differing observations on the same sample area would indicate that some change had occurred. Our above assertion should really have been that at a certain time on a certain date, fen was observed on area A. To save the repetitive specification of time, however, we shall assume that our information is static. This, together with the above assumptions, means that the same observation should always yield the same result over the same sample area.

1.3 Two Desiderata

Given that we are collecting information through observations to enable us to make statements about areas of the Earth's surface, we desire that the way we represent the information should allow two important capabilities:

- Since we cannot claim that the observations are complete or exhaustive, we should be able to incorporate subsequent observations into the earlier data; here we are thinking not of time evolution, but of progressively refining our knowledge by follow-up observations. (Update from observation).

- We will (often!) want information about an area other than the one directly sampled — for example, in searching for correlations between different features, or in assessing the environmental impact of a proposed development. Hence we must be able to make some kinds of statements about unsampled areas, based on the information available on sampled ones. (Extrapolation from observation).

2 A Simple Model

Let us construct a simple model to meet the above requirements. Let the set of all areas on the Earth's surface be called Area, and let $\mathcal{P}$ be a set of predicates, such that all the statements we wish to make about a given area can be expressed in the form $P(A)$, where $P \in \mathcal{P}$ and $A \in$ Area. For the moment, we may view an area as a connected set of points, using set intersection to represent the overlap of two areas, set union their total coverage, etc. We will always assume that the areas formed in these 'areas expressions' turn out to be areas in the sense of being connected.

Use of expressions like $P(A)$ can be viewed as a simple extension of predicate calculus, meaning 'the statement P is true over the area A'. In this framework our correctness of observation assumption may be expressed as (1). Here P is the feature predicate, and $\mathrm{obsv}(P, A)$ means that P was observed to be true of A; that is, it is the observation predicate.

$$\forall P : \mathcal{P}, a : \mathrm{Area} \bullet \mathrm{obsv}(P, A) \Rightarrow P(A) \tag{1}$$

Let us define two classes or predicate, $\mathcal{P}_\exists$ and $\mathcal{P}_\forall$, such that predicates in $\mathcal{P}_\exists$ are true by the existence of (at least one example of) a feature, and those in $\mathcal{P}_\forall$ by the universality of a feature. An example from $\mathcal{P}_\exists$ might be that the sample area contains at least one specimen of a rare species; an example from $\mathcal{P}_\forall$ that the sample area is (entirely) fenny.

(Features of the class $\mathcal{P}_\exists$ are important in conservation work, where the motivation for protecting a certain site may be the presence of a small number of rare specimens, or even a single instance; however, such marginalities are often poorly handled by statistical data management techniques).

We shall explore how these update and extrapolation *desiderata* can be provided over these two classes of predicate. Later more general classes of predicate will be considered, and a more rigorous treatment of the idea of 'trueness over an area' given.

2.1 A Calculus of Areas and Predicates

Let us extend our intuitive concept of trueness over an area by considering the following inference rules; for $P \in \mathcal{P}_\forall$, $A \in$ Area, and $B \in$ Area,

$$P(A) \wedge B \subseteq A \Rightarrow P(B) \tag{2I}$$

$$P(A) \wedge P(B) \Rightarrow P(A \cup B) \tag{2II}$$

$$P(A) \Rightarrow P(A \backslash B) \tag{2III}$$

All of these may stand on appeals to intuition. (I) says that if something is universally true of some area, it is also true of any subarea; (II) says that a statement true of all of two areas is also true of all of the area formed by grouping together these two areas, if this is possible (ie, the first two areas are adjacent or overlap, so they group to form a single connected area). (III) says that a statement true of all of an area is true of the area with any part removed.

Note that we may consider $P(\emptyset)$, where $\emptyset$ is is the empty area, to be a null assertion, since it contains no reference to a finite area over which the predicate is true. For this reason we may assert (3); this is an extension of the 'excluded middle' law of classical logic.

$$P(A) \wedge \neg P(A) \Rightarrow A = \emptyset \tag{3}$$

Now let us consider similar inference rules where $P \in \mathcal{P}_\exists$, $A \in$ Area, and $B \in$ Area.

$$P(A) \wedge A \subseteq B \Rightarrow P(B) \tag{4I}$$

$$P(A) \Rightarrow P(A \cup B) \tag{4II}$$

(I) states that something to be found in an area A is also to be found in area B if area B contains area A. (II) states that something to be found in A is also to be found in any area of which A is a part. We may view (4II) as a corollary of (4I) since it is easily derivable from it.

2.2 Update from Observation

We require that the model allow update of our knowledge of the world in the light of subsequent observation. Moreover, we wish to derive as much information as possible from subsequent observations — that is, we wish to make the strongest statements possible which the combined observations will validate.

For example, if $P \in \mathcal{P}_\forall$, and we make two observations, $\mathrm{obsv}(P, A)$ and $\mathrm{obsv}(P, B)$. By assumption (1) we may assert $P(A)$ after the first observation, and $P(B)$ after the second. However, using (2II) we now assert $P(A \cup B)$. This is the strongest statement we can make, since P says something is true of all of an area, and $A \cup B$ is the biggest area for which we can validate this statement.

Now, if $P \in \mathcal{P}_\exists$, and we make the same two observations, we can again assert $P(A \cup B)$, by (4II).

However, if A and B overlap, we may assert $P(A \cap B)$ in the first case but not in the second. (Note that $P(A \cap B) \in P(A \cup B)$, and note the different forms of (2I) and (4I)). That is to say, if something is true of all of two areas, then it is true of their overlap if it exists; but if we can find an example of something in area A, and also find one in area B, we can not necessarily find one in their overlap if it exists.

2.3 Extrapolation from Observation

We have considered how an observation validates a statement about the sample area in which it was made; however, we wish to be able to make statements about areas which are not sample areas, and to be able to assess the degree of confidence with which these statements may be made.

Let us assume that we have made an observation on area A, validating predicate P. How does P apply to a new area B, which may stand in arbitrary geometrical relation to A?

First let us consider the case where $P \in \mathcal{P}_\exists$; here we may assert:

$$P(A) \Rightarrow \pi P(B) \geq \mid A \cap B \mid / \mid A \mid \tag{5}$$

Here $\pi P(B)$ is the probability that P is true of B; we have assumed that the specimen to which P refers is as likely to be in any part of A as any other. The modulus signs are used to mean the surface area (measure) of a given area. This expression applies irrespective of the geometrical relationship of A to B. If A and B are disjoint, we have no information on B. If A is included in B, we know that P is true of B. Otherwise, we have a minimum probability that it might be true.

Now let us consider the case where $P \in \mathcal{P}_\forall$; here, we may assert:

$$P(A) \Rightarrow \rho P(B) \geq \mid A \cap B \mid / \mid B \mid \tag{6}$$

Here $\rho P(B)$ means the fraction of B to which we know P applies, by virtue of its inclusion in A. Without any knowledge of the distribution of this feature, we cannot assess the likelihood of its presence in the remainder of B. Again, the expression applies irrespective of the geometrical relationship of A to B; in both cases, we have assumed an 'even distribution' of the feature property.

3 Atomisation

We have shown how a calculus of areas might be used to attain the *desiderata* of section 1.3. We have considered only two classes of predicate; although this approach has been extended to more general examples (for example, involving quantification) the results become increasingly complicated. More importantly, we note that the rules by which we might make updates and perform extrapolations are not independent of the semantics of the feature predicates which we are considering.

Hence *atomisation* is introduced as a procedure for meeting our *desiderata* in the context of more general classes of predicate; this approach subsumes the calculus of areas and, further, allows conceptual modelling to be performed in an object oriented framework (see section 4.1). Let us define an atom to be an indivisible finite area of surface, the smallest area over which we shall make an observation or statement. It should be the smallest area referred to in the survey, or any survey with which we are likely to wish to combine our results. Let Atom be the set of all atoms over our survey area, and Area be the set of all areas formable from them. An area is then a set of any atoms which are connected (7). By *connected,* we mean that there is a path from any atom in the area to any other which does not pass outside the area. (Testing this for a given area would require a graph traversal).

$$\text{Area} = \{A : \mathbf{P}\text{Atom} \mid a \in A \wedge b \in A \Rightarrow \text{connected}(a, b)\} \tag{7}$$

where **P**Atom is the power set of Atom.

Such an approach seems justified since in general the methods of surveys are finitary. Since atoms are of finite size, we may use the cardinality of an area to refer to the number of atoms which it contains.

We require that a statement is either true or false of an atom; it is never partially true. This allows $P(a)$ to be interpreted as in classical predicate logic, where $a \in$ Atom. Our earlier statements of the form $P(A)$, where A is an Area, may now be interpreted as follows.

$$P(A) \equiv \forall a : A \bullet P(a), \text{ for } P \in \mathcal{P}_{\forall} \tag{8I}$$

$$P(A) \equiv \exists a : A \bullet P(a), \text{ for } P \in \mathcal{P}_{\exists} \tag{8II}$$

In other words, in (I) the distributed conjunction of the predicate is true over all atoms in the area, and in (II) the distributed disjunction is true. This gives us a more formal statement of our concept of 'trueness over an area'. Further, we may now deduce the laws of section 2.1 using only classical predicate logic and (8I) and (8II) — see the appendix for an example.

4 Object Oriented Survey Data

In this section, an object oriented model of survey data is developed.

4.1 Object Oriented Conceptual Modelling

Conceptual modelling [BMS84] is a tool for the high-level description of data. Object oriented modelling [BGHS91] is intended to provide strong semantic capture, using the underlying metaphor of a physical system [MMP91]. It provides means to model composite objects, and further to capture behaviour or calculation within the model; this latter feature is seen as one of the distinguishing features of an object oriented approach [Kin89]. Work such as [WHM90] has used the structural properties of object oriented models to represent spatial data; here we will use also the behavioral aspects, to represent the derivation of properties.

For brevity, basic concepts of object oriented modelling are not reviewed here; such a review may be found in [BK91], together with a presentation of the model used in this article. Here we introduce relevant features as they occur in examples.

4.2 Equality of Areas

Before approaching the model, let us detour briefly to look at the concept of equality over areas. Two forms of equality are of interest when discussing areas, which must be distinguished as discussed in [BK91]. Two objects of class Area may be the same object, or they may be the same area. For example, if there were a development site in the survey site, this might be represented by an object of class DevelopmentSite. If the developers buy a little more land abutting the site, but are refused planning permission for one corner of the original site, so that the shape of the DevelopmentSite alters, then we call this the same object as before, although not the same Area since it is not the same set of Atoms (shallow equality); this is the default equality method inherited from class Object.

Now let us imagine that we query the database for the habitat of a certain type of vegetation; the reply will be the set of atoms where it is located. If these are adjacent, we may regard it as a temporary area object created in reply to the query. If this area represents the same piece of ground (is the same set of atoms) as some existing feature, for example a bog, then the habitat and the bog are the same area, but not the same object (deep equality). For example, an increase in the size of the bog does not necessarily mean an increase in the size of the area where the vegetation is located.

To express this we consider that in addition to the object equality inherited from class Object, Area and its subclasses have a method `areaEqual` which will show whether two objects in fact comprise of the same terrain.

4.3 SampleAreas, InterestAreas and their Properties

In a survey, the observations made are vital. Any model which prevents the storage of these actual observations will be undesirable, since they are the basic and incontestable results of the survey. Derived data might always need to be rederived, since we may change our minds about which features we are interested in; this is possible only if the original observations are still available. Therefore any model should preserve the data actually observed, while simultaneously allowing us to see it in different ways as we might wish.

In our discussion of survey data we have mentioned areas on which observations have been made, areas over which we wish to make statements, and we have introduced the notion of an atom as a mechanism for connecting the two. We capture this in the following model (see figure 1). (To reduce clutter, natural language comments between braces replace some elements in the accompanying schemata). The diagram shows that an area is composed of one or more atoms, and that Area has two subclasses, SampleArea and InterestArea. A SampleArea is an area on which observations have been made, whereas an InterestArea is an area about which we wish to make statements. Of course, any given area of interest may have been sampled; so here we restrict InterestArea to mean an area about which we wish to be able to make statements. This may or may not be `areaEqual` to some SampleArea. The properties of these subclasses of Area represent the features of their real-world counterparts in which we are interested.

The notation used is as described in [BK91]. Fat arrows are used to show the generalisation structure of the model, and thin arrows to show the aggregation structure [SS77]. The arrow `atoms` shows that an area consists of a number of atoms. The constraint bar labelled `CC` (connectivity constraint) means that the constituent atoms must form a connected aggregation as described in section 3. The fat arrows show that SampleArea and InterestArea are both subclasses of Area; hence objects of either of these two classes also consist of connected sets of atoms. Further, these classes inherit both methods for equality testing from their common superclass.

Depending on the survey methodology used, any particular application model may involve various different kinds (subclasses) of SampleArea. Similarly, depending on how the data is being used, various subclasses of InterestArea may also exist in any given application.

4.4 Atomisation Revisited — Decomposition and Recomposition

The properties of a SampleArea represent the features which we observe, those of an InterestArea represent statements which we may make; we may use atomisation as a bridge between the two.

We may decompose the value of a feature of a SampleArea to a value for each of the constituent atoms, and then recompose these into a value for any InterestArea including these atoms (see figure 2). The rule for decomposition and recomposition will depend on the feature predicate concerned.

This approach subsumes the calculus of areas, since the latter may be interpreted in terms of the atomisation represented by (8I) and (8II), where simple decomposition and recomposition rules apply.

It will still be possible to meet the two *desiderata* in more general cases, provided that appropriate decomposition and recomposition rules can be found, since updating by observation simply means that more atoms must be operated on by the decomposition rule, and recomposition is itself a process of extrapolating from observation. Since the value of a property of an object may be found by evaluating a method body, the decomposition and recomposition may be done (at least notionally) at query time. The composition rules are represented by the specification of the property in the schema for InterestArea, utilising the capture of computation provided by object oriented models.

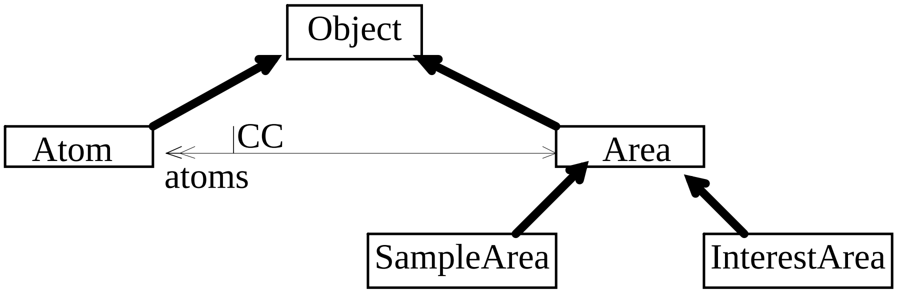
4.5 Correlated Features

To represent statements we may wish to make about arbitrary areas, we introduce a new feature as a property of the InterestArea. This is derived from the actual feature of the SampleArea, and is called the *correlated feature.* The correlated feature has a value for any arbitrary areas of interest; where the InterestArea is the same area as some SampleArea, the value of the correlated feature is reducible to that of the (observed) feature from which it is derived. This is the *correlation constraint* between the feature and its correlated feature. We have already tacitly adopted this approach while trying to extrapolate from observation in section 2.3, where we introduced probabilities and proportions.

Several different correlated features might be based on the same actual feature; also, the same subclass of InterestArea might have as properties correlated features based on a variety of classes of SampleAreas.

4.6 Example

Let us consider the following example. Imagine that a survey methodology were to look for fenny patches on a survey site, and to chart them. We would then have a class FennyPatch, a subclass of SampleArea.



```
class Area
properties
  atoms: setOf Atom                      ;;
operation
  equalArea: Area other -> Bool is
       self.atoms.setEqual(other.atoms)   ;;
constraint
  CC: {atoms connected}                  ;;


class SampleArea
ISA Area
properties
  {feature properties}
  ----- ;;
  ----- ;;
  ----- ;;

class InterestArea
ISA Area
  {correlated feature properties}
  ----- ;;
  ----- ;;
  ----- ;;
```

Figure 1: Ecological Data Model

Figure 2: Atomisation

Now, let us imagine that we wish to make statements about how fenny various arbitrary parts of the site might be. Therefore, we can give the class InterestArea a property `minPercentageFen`, showing what proportion of an area we know to be covered in fen. We represent this as a minimum, since we consider that the survey site may be incompletely surveyed, so that we might lack knowledge of some of the fen existing within our InterestArea. `minPercentageFen` is the correlated feature to the property `fenny` implicitly possessed by objects of class FennyPatch. The methods are expressed in an extension of NOODL [Bar92], which is derived from the notation used in [BK91]. The specification of the property `minPercentageFen` is

```
(accumulate(FennyPatch, self.intersect)).cardinality  / (self.cardinality)
```

So this method means: divide the number of atoms in the intersection of the InterestArea with all known fenny patches by the number of atoms in the InterestArea. This gives the minimum proportion of the InterestAreas known to be fenny (see figure 3). If we had exhaustively surveyed the site for fenny patches, this would be the exact proportion which was fenny.

Now, consider the case where the InterestArea is `areaEqual` to a fenny patch; that is, the InterestArea covers the same terrain as an object in the class FennyPatch. In this case, the intersection of the InterestArea with all known fenny patches `areaEquals` the InterestArea itself, so the method simplifies to `self.cardinality` divided by `self.cardinality` which will be 1 as we expect. Thus we are able to recover the information from our original observation. The same argument will hold if the area of interest is a subarea of any fenny patch. This is the correlation constraint for this feature. The schemata for these classes are shown in figure 4; the de- and re-composition rules are implicit in the method which defines the correlated feature.

One advantage of this approach is its flexibility. We can choose which properties an InterestArea should have, based on the available observations on SampleAreas. Instead of the property `minPercentageFen,` we could give the InterestAreas a Boolean-valued property `fenny.` The associated method would then check whether the SampleArea was `areaEqual` to some object of class FennyPatch. Alternatively, we could weaken the definition slightly and check whether it overlapped 90% or more with an object of class FennyPatch. We could alter the tolerance, indeed the whole approach, till we found what is most useful in a given case. Of course, InterestArea could have all of these different correlated features simultaneously, provided the properties by which they are represented had different names.

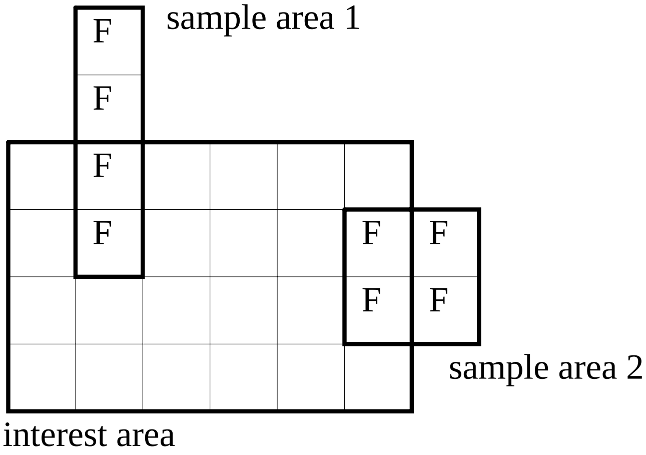


Figure 3: Example

```
class FennyPatch
ISA Area
constraint
  isFenny: {all atoms fenny} ;;

class InterestArea
ISA Area
  minPercentageFen: Proportion is
     (accumulate(FennyPatch, self.intersect)).cardinality / self.cardinality ;;
```

Figure 4: Example Schema

At the cost of increasing complexity and decreasing certainty, we could base the properties of an InterestArea on quantified observations of SampleAreas, calculating the probability that each atom of the SampleArea contributes to the desired property of the InterestArea.

4.7 Atomic SampleAreas

A simplification which might be enabled by the methodology of many surveys is to work only with atomic SampleAreas; that is, SampleAreas which consist of a single atom only. In this case, the decomposition step may be omitted, since the predicate true of the SampleArea is also true of its constituent atom. Survey methodologies are such that this simplification is often possible.

To recast the above example with this simplification, we might imagine that the survey methodology was to record whether atomic sample areas (perhaps small quadrats) were fenny; each atom might have a Boolean valued property `fenny`, or a discrete valued property `groundType,` with `‘fenny’` being one of its possible values. The method for the property `minPercentageFen` would then be replaced by:

```
(self.atoms where (self.fenny)).cardinality / self.cardinality
```

or

```
(self.atoms where (self.groundType = ‘fenny’)).cardinality / self.cardinality
```

since we can attribute the observation-property to the atoms themselves. That is, we divide the number of fenny atoms in the SampleArea by the total number of atoms in the SampleArea.

The simplification obtained here is fairly limited since the decomposition procedure for the original example was simple: an atom in a fenny patch is a fenny atom. However, it is easy to find cases where the decomposition procedure is considerably more complicated, especially where observations involve quantification, and/or SampleAreas overlap. In these cases decomposition may require involved probability calculations and assumptions about how a feature is distributed in order to allocate a value to each of the atoms in the SampleAreas.

The consequence of the use of atomic SampleAreas is that we may make only discrete valued observations of SampleAreas, since the corresponding proposition must be simply either true or false of the atom (see section 3); this would reflect a methodology where we must decide to allocate the quadrat to one of a number of predefined categories.

4.8 Review of the Model

We have represented areas of the Earth's surface as objects, and features of interest by properties of these objects. We have made use of the 'stored query' view of a property in order to give meaning to these properties for any of the infinite number of different area objects with which we might wish to work.

The model presented captures the essential requirements identified in section 1.3. We capture the need to make statements about areas other than those directly observed by the introduction of the two classes SampleArea and InterestArea. All observations are made on objects of class SampleArea, all statements about objects of class InterestArea. The correlation constraint guarantees the original observations are recoverable when the InterestArea is `areaEqual` to a SampleArea.

The requirement of updatability is also met since the features of InterestAreas are expressed in terms of a method upon the observed properties of SampleAreas. Hence implicitly, an alteration to the observed data is reflected in the features of any InterestAreas. (In a stored database, this might mean either that the values of these properties are evaluated dynamically, or that they are precomputed, but recomputed on update of the relevant observations).

Further, the object oriented approach has proved useful, not only for capturing the composite structure of data objects, but also for allowing the necessary derivations to be expressed within the data model.

5 Paisley Data

The approach discussed above is being used to model ecological survey data collected in a survey undertaken by Paisley College of Technology, in conjunction with the Nature Conservancy Council. Here we give a brief overview of the structure of a subset of the data; more details may be found in [BCM88].

The survey was carried out in 1986 and 1987, and has a fourfold purpose:

- To establish the distribution and numbers on Islay of birds listed in EC directive 85/411/EEC Annex 1.
- To produce a land type classification of the island.
- To investigate the relationship between land use and birds to assess the potential impact of changes.
- To provide recommendations to incorporate into the development of a conservation strategy to maintain and enhance the wild-life interest of the island.

In this examination we consider only the data relating to land classification, although it is planned to extend this later to the bird data.

A sample of 1km × 1km quadrats was selected, and each divided into 2500 50m × 50m squares (called *pixels* in the report). These were surveyed for a variety of biotic features, such as woodland and vegetation types present, and abiotic features such as physiography and boundary features. This information was collected on coding sheets, using a different coding scheme for each square surveyed. Subsequently, the data was reduced to a standardised representation scheme and subjected to TWINSPAN (Two Way INdicator SPecies ANalysis) [Hil79], to determine various categories for the different features of interest. This procedure is described fully in [BCM88].

5.1 Brief Overview of the Data

Our aim in modelling the data is not to redo the classificatory analysis already performed upon it, but rather to be able to represent the results of this initial work in an intuitive manner, allowing graphical display of the data, and *ad hoc* querying on it. Any conclusions derived from the data must be supported by the rigorous statistical techniques employed by ecologists; we hope to offer a facility to browse the data in order to perform explorative 'searching for patterns.'

Analysis revealed the pixels surveyed fell into eight boundary categories (`B`), nine physiographic categories (`P`), four woodland categories (`W`), and twenty-six agriculture and vegetation categories (`AV`). Sets of contiguous pixels of the same type form *patches*, which are naturally occurring areas homogeneous in some feature.

Further, the 1 km × 1 km squares were categorised into eight landtypes according to the frequencies of various patch types occurring in each. Unsampled squares were also categorised on the basis of map information. The entire island can thus be divided into *zones* which are large areas characterised by considerable homogeneity of landtype.

5.2 Description in Terms of Model

The decision to use the 50m × 50m pixels as sampling areas was intended to support a standard survey methodology [BS73] allowing follow-up surveys and comparison with other surveys over the same site. Additionally, it means that we have the advantage of atomic sample areas as discussed in 4.7. We can treat the pixels, squares, patches and zones as objects, and their landtype characteristics as properties. The structure of the model derived is shown in figure 5, together with some of the associated schemata (constraint bars and labels are omitted for clarity). Pixels and Quadrats are our basic forms of SampleArea, and Patches, Zones and Mixels (see section 5.4) the appropriate subclasses of InterestArea. Since Pixels are the only kind of atoms, we use the class Pixel as a synonym for Atom in this application.

5.3 Querying

A graphical user interface has been constructed for use with the ecological database (see figure 9). A likely class of queries would be representational in nature - for example, display all pixels of whose `typeAV` is `b` and whose `typeW` is `d.` The query would involve locating all pixels which matched the selection criterion, after which they would be passed as a set of objects of class Pixel to the graphical user interface for display in an appropriate manner. If these pixels were of some interest, we might wish dynamically to create a new subclass of Pixel, SpecialPixel, as defined in figure 6, so that they might persist for use in future querying sessions.

A second class of query might be to indicate some area on the screen by highlighting it with the mouse, and then examine its features treating it as an InterestArea. Clearly this would involve dynamic evaluation of its correlated features. Should the area be of interest, it might be added to the class UserDefined for later reference.

5.4 Mixels

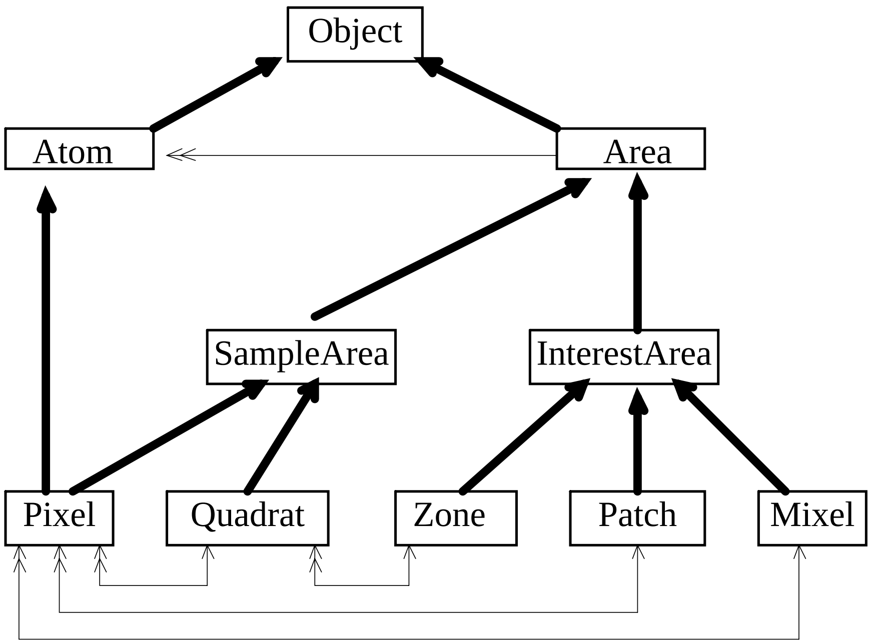
An interesting example of using InterestAreas which are not SampleAreas is the creation of *mixels.* A mixel (= 'mixed pixel') is a square set of pixels, whose characteristics are derived from those of the constituent pixels. The creation of mixels allows us to increase the coarseness of granularity with which data is treated. This might for example be useful when searching for correlation between the locations of a species and certain patch types; if the species is observed near the limits of its preferred habitat, we might overlook the correlation if working at too fine a level of granularity. Of course, the rules by which the characteristics of the mixel are derived from those of the pixels are not necessarily clear, so that some experimentation might be necessary to find meaningful ways of constructing mixels.

For the sake of an example, we shall use a very simple approach to deriving our mixels. Let us consider a class Mixel4, the members of which are mixels consisting of four adjacent pixels arranged in a 2 × 2 square. If all four of the constituent pixels are of the same type, we shall declare the mixel to be of this type also; otherwise we assign it to a new type, `mixed.` Thus for each of the data sets, the number of mixel types is one more than the number of pixel types. (More complicated construction rules would, of course, introduce more types for various possible pixel combinations).

We can now introduce a new class Mixel4, a subclass of InterestArea, whose schemata is as shown in figure 7. For browsing a database, we might wish to be able temporarily to create objects of class Mixel4, using this and other definitions, and perhaps then permanently to store the objects created according to some of the definitions if they appear to form a useful basis for further work.

6 Implementation

A system called *Isis* (Islay Survey Information System) has been built to implement the application model described here; it is written in the persistent programming language Napier88 [DCBM89], [MBCD89], which



```
domain Index is Number where self >= 1
                       and   self <= 50

domain PatchTypeAV = ( {various discrete values} )
domain PatchTypeW  = ( {various discrete values} )
domain PatchTypeB  = ( {various discrete values} )
domain PatchTypeP  = ( {various discrete values} )

domain LandType = ( {various discrete values} )


class Pixel
ISA Atom { is same as Atom in this application }
   quadrat: Quadrat
            \ pixels         ;;
   xIndex:  Index            ;;
   yIndex:  Index            ;;
   patchTypeAV: PatchTypeAV  ;;
   patchTypeW:  PatchTypeW   ;;
   patchTypeB:  PatchTypeB   ;;
   patchTypeP:  PatchTypeP   ;;


class Quadrat
ISA Area
   xRef: Number                  ;;
   yRef: Number                  ;;
   landType : LandType           ;;
   pixels   : setOf Pixel
              \ quadrat          ;;
constraint
  igc: {implicit geometry of pixels}  ;;
```

Figure 5: Application Model

```
class SpecialPixel
ISA Pixel
constraint
  sc: self.typeAV = b and self.typeW = d  ;;
```

Figure 6: Special Pixel Schema

```
domain MixedMixelType is ( mixed )
domain MixelTypeAV is TypeAV or MixedMixelType
domain MixelTypeW  is TypeW  or MixedMixelType
domain MixelTypeB  is TypeB  or MixedMixelType
domain MixelTypeP  is TypeP  or MixedMixelType

class Mixel4
ISA InterestArea
properties
  tl: Pixel ;;
  tr: Pixel ;;
  bl: Pixel ;;
  br: Pixel ;;
  mixelTypeAV: MixelTypeAV is
      if tl.typeAV = tr.typeAV and
         tr.typeAV = bl.typeAV and
         bl.typeAV = br.typeAV then
      tl.typeAV
         else
      mixed                                        ;;
  {similar for W, P, B}
constraint
  mgc: {pixels in correct geometrical arrangement} ;;
```

Figure 7: Mixel Schema

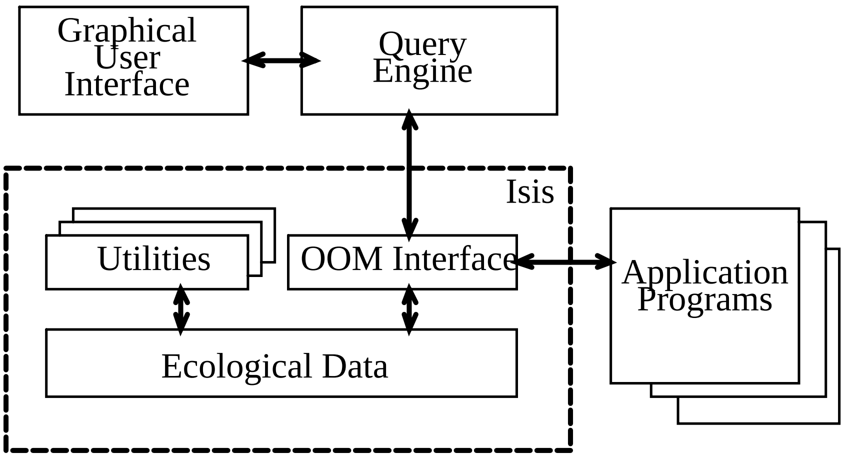


Figure 8: System Architecture

transparently manages the long-term storage of the data. The architecture of the system is shown in figure 8. *Isis* itself consists of persistent data, some utility applications which may access it directly, and an interface supporting the object oriented application model through which all other applications must access the data. The graphical user interface shown in figure 9 [BFK92], [Fra91] accesses the database through an intermediate query engine which translates the pictorial representation of a query into terms of the object oriented model. The graphical presentation of the data is thus separated totally from its representation in the database.

7 Summary

The requirements for a model for ecological data have been discussed. Atomisation has been introduced as a technique allowing these requirements to be met. By incorporating this atomisation in property-definition methods, an object oriented conceptual model for this type of data has been developed. An application model for a particular survey has been built on this approach, and this application model implemented in a data management system.

8 Further Work

The approach to modelling ecological data requires extension to incorporate time variation. It is planned to model more data, including some bird data, which should add some new interesting structure to the model. We plan also to investigate the capabilities of various existing object oriented database systems for representing this data, as well as developing our own persistent database and the graphical user interface further.

9 Acknowledgements

We wish to thank Professor David Curtis of Paisley College of Technology for his collaboration in this work, and for making available to us some of his data; and Colin Fraser, of Napier University, for the screendump of his graphical user interface system.

References

[Abd91] Abdulhakim A Abdallah. *The Design and Implementation of a Prototype Geographic Information System Using a Novel Architecture Based on PS-algol.* PhD thesis, University of Glasgow, May 1991.

[Bar92] Peter J Barclay. The NOODL Guide. Technical report, Napier University, Edinburgh, 1992. (In preparation).

Figure 9: Graphical User Interface to *Isis*

[BCM88] EM Bignal, DJ Curtis, and JL Matthews. Islay: Land Types, Bird Habitats, and Nature Conservation. Technical report, Paisley College of Technology, 1988.

[BFK92] Peter J Barclay, Colin M Fraser, and Jessie B Kennedy. Customised Interfaces in Persistent Environments. In Richard Cooper, editor, *proc 1st International Workshop on Interfaces to Database Systems*, Glasgow, 1992. Springer Verlag.

[BGHS91] Gordon Blair, John Gallagher, David Hutchison, and Doug Shepherd. *Object Oriented Languages, Systems and Applications.* Pitman, 1991.

[BK91] Peter J Barclay and Jessie B Kennedy. Regaining the Conceptual Level in Object Oriented Data Modelling. In *proc BNCOD-9*, Wolverhampton, Jun 1991. Butterworths.

[BMS84] Michael L Brodie, John Mylopoulos, and Joachim W Scmidt. *On Conceptual Modelling - Perspectives from Artificial Intelligence, Databases and Programming Languages.* Springer Verlag, 1984.

[BS73] RGH Bunce and MW Shaw. A Standardised Procedure for Ecological Survey. *Journal of Environmental Management*, 1(4):239 – 258, 1973.

[DCBM89] Alan Dearle, Richard Connor, Fred Brown, and Ron Morrison. Napier88 - A Database Programming Language? In *proc DBPL 2*, 1989.

[Fra91] Colin M Fraser. Persistent Systems for Graphical Interface Construction. Technical report, Napier University, Edinburgh, May 1991.

[Gau82] HG Gauch. *Multivariate Analysis in Community Ecology.* Cambridge University Press, 1982.

[Hil79] MO Hill. *TWINSPAN - a FORTRAN Program for Arranging Multivariate Data in an Ordered Two-Way Table by Classification of the Individuals and Attributes.* Section of Ecology and Systematics, Cornell University, New York, Jul 1979.

[Ken85] Jessie B Kennedy. A Study of Ecological Database Management and Associated Data Analysis. Master's thesis, Paisley College of Technology, 1985.

[Kin89] Roger King. My Cat is Object Oriented. In W Kim and FH Lochovsky, editors, *Object Oriented Concepts, Databases, and Applications.* Addison Wesley, 1989.

[MBCD89] R Morrison, F Brown, R Connor, and A Dearle. The Napier88 Reference Manual. Technical report, Universities of Glasgow and St Andrews, Jul 1989.

[MMP91] Ole Lehrmann Madsen and Birger Moller-Pedersen. Basic Principles of the Beta Programming Language. In Gordon Blair, John Gallagher, David Hutchison, and Doug Shepherd, editors, *Object Oriented Languages, Systems and Applications.* Pitman, 1991.

[Pie84] EC Pielou. *The Interpretation of Ecological Data: a Primer on Classification and Ordination.* John Wiley and Sons, 1984.

[RGHH91] S A Roberts, M N Gahegan, J Hogg, and B Hoyle. Application of Object-Oriented Databases to Geographic Information Systems. *Information and Software Technology*, 33(1):38 – 46, Jan/Feb 1991.

[SS77] JM Smith and DCP Smith. Data Abstractions - Aggregation and Generalisation. *ACM TODS*, 2(2):105 – 133, Jun 1977.

[WHM90] Michael F Worboys, Hilary M Hearnshaw, and David J Maguire. Object-Oriented Data Modelling for Spatial Databases. *International Journal of Geographical Information Systems*, 4(4):369 – 383, 1990.

[Wor92] Michael F Worboys. A Generic Object Model for Planar Geographic Data. Technical report, University of Keele, Feb 1992.

Appendix

Here a semi-formal proof of result (6) is presented, as an example of how the concept of atomisation allows the derivation of the inference rules used in the calculus of areas. Hence, this calculus is reduced to conventional predicate logic.

Let us note first, that assuming all atoms are of the same size,

$$\mid A \mid = \# A \mid a \mid \qquad \text{(lemma 1)}$$

We shall also use the well-known result

$$P \Rightarrow Q \equiv \neg P \vee Q \qquad \text{(lemma 2)}$$

Then for $P \in \mathcal{P}_{\forall}$,

i)	$P(A)$	premiss
ii)	$\forall a : A \bullet P(a)$	by (8I)
iii)	$a \in A \Rightarrow P(a)$ for any atom a	by $\forall$-elim
iv)	$\neg(a \in A) \vee P(a)$	by lemma 2
v)	$\neg(a \in A) \vee \neg(a \in B) \vee P(a)$ for any area B	by $\vee$-intro
vi)	$\neg(a \in A \wedge a \in B) \vee P(a)$	by DeMorgan
vii)	$a \in A \wedge a \in B \Rightarrow P(a)$	by lemma 2
viii)	$a \in (A \cap B) \Rightarrow P(a)$	by $\cap$-defn
ix)	$\forall a : (A \cap B) \bullet P(a)$	by $\forall$-intro
x)	$P(A \cap B)$	by (8I)

Since $(A \cap B) \subseteq B$, the number of true atoms in B is at least $\#(A \cap B)$. Hence $\rho P(B)$, the proportion of true atoms in B, is given by:

xi)	$\rho P(B) \geq \#(A \cap B) \;/\; \# B$	by ρ-defn
xii)	$\rho P(B) \geq \mid A \cap B \mid / \mid B \mid$	by lemma 1

QED