
Spatial Cognition and Computation **2:** 261–282, 2000.

Spatial data quality capture through inductive learning

MATT DUCKHAM[1], JANE DRUMMOND[2] and DAVID FORREST[2]
[1]*Department of Computer Science, University of Keele, ST5 5BG, UK (Fax: +44 1782 713082; E-mail: m.duckham@computer.org);* [2]*Department of Geography and Topographic Science, University of Glasgow, G12 8QQ, UK*

Received 16 November 2000; accepted 10 December 2001

Abstract. The relatively weak uptake of spatial error handling capabilities by commercial GIS companies and users can in part be attributed to the relatively low availability and high costs of spatial data quality information. Based on the well established artificial intelligence technique of induction, this paper charts the development of an automated quality capture tool. By learning from example, the tool makes very efficient use of scarce spatial data quality information, so helping to minimise the cost and maximise availability of data quality. The example application of the tool to a telecommunications legacy data capture project indicates the practicality and potential value of the approach.

Key words: data quality, GIS, inductive learning, information content, object calculus

1. Introduction

The term *error-sensitive GIS* (Uwin 1995) refers to the concept of a GIS capable of handling both geographic information (GI) and the uncertainty that inevitably surrounds GI. Work on the development of error-sensitive GIS has progressed on a number of fronts, including the management of information about error (e.g. Ramlal and Drummond 1992; Duckham and Drummond 1999; Qiu and Hunter 1999), error propagation (e.g. Lanter and Veregin 1992; Openshaw et al. 1991; Wesseling and Heuvelink 1993; Heuvelink 1998), and the development of error-sensitive user interfaces and visualisation techniques (e.g. van Elzakker et al. 1992; van der Wel et al. 1994; Agumya and Hunter 1997; Hunter 1999; Bastin et al. 1999). Despite increasing awareness amongst GIS companies and users of the importance of data quality, very often adequate data quality information for a data set will simply not exist. Further, limited expertise and financial restrictions are likely to mean most data producers may not feel in a position to compile such quality information about their geospatial data. Veregin (1989) places the identification and assessment of errors at the root of a hierarchy of needs for effective error-

handling in GIS. There exists, therefore, a clear need for error-sensitive GIS tools that assist not simply in management, propagation and visualisation of information about uncertainty in GI but additionally to assist with the capture and production of data quality.

Some attempts to address this need have already appeared in the literature. Lanter (1991) describes a meta-data system capable of tracking the development of GI and automatically producing appropriate lineage information. The technique of least-squares adjustment (Mikhail 1978) has been proposed as a basis for a *measurement-based GIS* (M-BGIS) capable of automatically producing positional accuracy information based on original survey data (Campbell et al. 1994; Goodchild 1999). However, such attempts are both scarce and relatively specialised, such that there exist no general purpose tools that can claim to automate the capture and production of spatial data quality information. By developing automatic data quality capture systems, one of the major barriers to practical error-sensitive GIS deployment may be overcome, making the use of error-sensitive GIS a viable option in a wider variety of applications than currently feasible. In the context of this situation, this paper explores the use of an inductive learning algorithm as a basis for flexible and automatic capture of data quality information in GIS.

2. Background

This section reviews three important elements in the approach to automatic data quality capture proposed here. First, the need for automatic data quality capture is demonstrated by introducing the example of a telecommunications application. Second, the use of inductive learning algorithms is proposed as a mechanism for achieving automatic data quality capture. Third, the role of object-orientation (OO) in the development of automatic data quality capture is reviewed.

2.1. *Telecommunications and data quality*

The specific application considered by this research concerned the migration of telecommunications network plans to digital mapping. In 1997, Kingston Communications (KC), Informed Solutions and Survey and Development Services (SDS) undertook the capture of the entire telecommunications network for Kingston-upon-Hull, UK, within an OO GIS. Prior to 1997, spatial data management at KC had relied primarily on Ordnance Survey of Great Britain (OSGB) 1:1250 base maps with telecommunications features marked on by hand. The migration away from paper towards digital mapping practices is a common feature of the deregulated UK telecommunications

industry, motivated largely by the improvements in logical and topological consistency afforded by GIS. Indeed experiences during this study suggested that there is, in fact, a high level of informal awareness of data quality issues amongst GIS professionals. For example, initiatives such as the National Land Information Service and the Scottish Land Information Service are adding momentum to the development of an integrated LIS in the UK (Smith 1996). Utility companies in particular are well placed to benefit from and contribute to such initiatives; generally such companies are aware that management of data quality may be a vital component of this increased integration.

Aside from logical and topological consistency, broader provision for data quality management is not usually a feature of data capture projects, such as that undertaken by KC. Lack of expertise, wariness of relatively new error-sensitive technology and negative connotations of error will all be considerations that militate against long-term data quality management. Such considerations are both causes and symptoms of a self-perpetuating cycle of under-investment in data quality. Arguably, it is the high levels of investment needed to perform full quality assessments that underly this situation. In order to break the cycle simple, efficient and cost effective methods of data quality capture are required.

2.2. *Inductive learning algorithms and data quality*

In order to address the needs of the telecommunications industry and offer automated assistance with data quality capture, this research took advantage of a powerful artificial intelligence (AI) technique used for automated learning from example: *inductive learning*. Given a training data set, an inductive learning algorithm should be able to automatically deduce rules that embody the patterns in that data, rules which, hopefully, correspond to underlying processes governing the data. Quinlan (1979) describes the ID3 inductive learning algorithm in its application to the chess endgame. Given a suitable set of example endgame positions, Quinlan's ID3 algorithm was able to induce a set of rules that describe these examples. Having undergone this training, the induced rules can then be applied to chess endgames more generally, even endgame situations that were not part of the original training set.

Inductive learning algorithms are not new to GIS. Walker and Moore (1988) use induction to identify relationships between geospatial objects and help with an automated habitat classification process. Similarly, Aspinall (1992) applied induction to the problems of habitat analysis, while Bennet and Armstrong (1996) use induction to assist with drainage feature extraction from a DEM. From the point of view of automatic data quality assess-

ment, inductive learning algorithms offer the possibility of making the most efficient use of the available data by constructing reasonable inferences from scarce data quality information. However, inductive learning algorithms still require adaptation before they are suitable for use with spatial data quality. Use of induction in GIS is not widespread as induction is primarily designed to deal with discrete categorical data, and so the technique requires modification to cope with the inherently continuous spatially referenced data commonly used in GIS. While data quality information is generally aspatial and does not rely too heavily on the spatial nature of the data to which it refers, the existence of spatial relationships, for example autocorrelation in errors provides additional complexity that needs to be overcome.

2.3. *Object calculus*

The issues covered by this paper are tackled from an object-oriented (OO) perspective. The superior semantic modelling capabilities of object-orientation, when compared with alternatives such as the relational data model, are well documented (see Egenhofer and Frank 1989; Worboys et al. 1990; Kösters et al. 1997) and OO should now be regarded as familiar part of the mainstream of GI technology. To facilitate a precise formal discussion of OO GI, this section briefly introduces an algebra for objects, called ς (sigma) calculus.

The ς-calculus (Abadi and Cardelli 1996) provides a simple and robust formalism with which to explore object systems. The ς-calculus can be used to model objects in the same way as the relational algebra (Codd 1970) is used to provide a formal model of relational databases or as λ-calculus (Hankin 1994) is used to provide a formal model of functional programming languages. The ς-calculus has already proved useful in the development of OO GIS (Duckham 2001). Briefly, an object in the ς-calculus is represented as a collection of named methods, l_i, each with method bodies b_i. The symbol ς is used to bind the postfixed 'self' parameter (conventionally s or z) with occurrences of that parameter in the body of the method, written $\varsigma(s)b_i$. Each object is enclosed in square brackets an associated with a label using the symbol $\triangleq$ (equal by definition), illustrated in Equation 1 below.

$$o \triangleq [l_i = \varsigma(s)b_i^{i \in 1 \ldots n}] \tag{1}$$

Invocation of a method l on an object o, written $o.l$, causes the body of the method l to be evaluated by substituting the object o for occurrences of the self parameter in the body of l. A full exposition of method invocation of ς-calculus objects (termed *reduction*) is not necessary here, particularly since the informal semantics of the reduction process will be familiar to anyone

used to object-oriented programming (OOP) techniques. Reduction can be illustrated informally with the *LineSegment* object in Equation 2 below. In Equation 2, the *LineSegment* object is described by two x,y coordinate pairs (0,0) and (3,4). Invocation of the *length* method on the *LineSegment* object reduces to 5 (written $LineSegment.length \rightarrowtail 5$) as expected.

$$LineSegment \triangleq [x_1 = \varsigma(s)0,\ y_1 = \varsigma(s)0,\ x_2 = \varsigma(s)3,\ y_2 = \varsigma(s)4,\ length = \varsigma(s)((s.x_1 - s.x_2)^2 + (s.y_1 - s.y_2)^2)^{\frac{1}{2}}] \quad (2)$$

Following from Equation 2, two further points are worth noting. First, the existence of natural numbers assumed in Equation 2 is a notational convenience and not part of the core ς-calculus. Second, where the bound self parameter $\varsigma(s)$ is unused in the body of the method it is conventionally omitted (e.g. $x_1 = \varsigma(s)0$ becomes $x_1 = 0$): such methods are usually termed attributes or fields.

3. Induction

All induction algorithms share a number of features in common. In essence, we can define induction as operating upon a set of (ς-calculus) objects $T = \{o|o \triangleq [l_k = a_k]^{k\in 1\ldots n}\}$ called the *training set*. Additionally, each object in the training set is classified as belonging to exactly one category C_i such that $C_i \subseteq T$ and $C_i \neq C_j \Rightarrow C_i \cap C_j = \varnothing$. An inductive algorithm is able to build a decision tree that embodies the data in the training set using the following three steps, after Quinlan (1983).

1. if the training set of objects is empty, $T = \varnothing$, we associate a new leaf in the decision tree arbitrarily with one of the categories C_i.
2. if all objects in the training set belong to the same category $T \subseteq C_i$ then we create a new leaf in the decision tree with that category C_i.
3. else we select an attribute l and partition T into disjoint sets $T_j^{j\in 1\ldots m}$ where T_j contains members with the jth value of the selected attribute, $T_j^{j\in 1\ldots m} = \{o|\forall o \in T \quad o.l \rightarrowtail x_j\}$. A new decision node is then created to represent this decision and the algorithm is reiterated using each subset T_j.

Even in this stripped-down form, the induction algorithm is surprisingly powerful and will always successfully categorise a set of objects, provided there are no two objects that have identical attribute values but belong to different categories (Quinlan 1983) – i.e. as long as the statement $\forall o_1 \in C_i, o_2 \in C_j \exists l\ o_1.l \neq o_2.l$ holds where $C_i \neq C_j$. When this condition does not hold, it indicates that there is not enough attribute information about objects in different categories to tell them apart. In reality this condition will

occasionally not hold and a practical inductive learning algorithm will usually need to resort to some heuristic, tackled in §4.3, to resolve such conflicts.

The actual performance of the inductive algorithm is dependent to a large extent on how the algorithm selects the attribute l with which to partition the set T in 3 above. There are a range of different methods that might be used to achieve this, but one of the most efficient is to use information theory. The mathematical concept of information theory was first defined by Claude Shannon in the late 1940s (Shannon 1948). Shannon's information theory formalises the *information content* of a statement in terms of a number of binary digits or *bits* of information conveyed by the statement. For example, when tossing a coin, the value of knowing the outcome has an information content of 1 bit. However, if it is already known that the coin is biased, the value of knowing the *actual* outcome is reduced. The amount by which the value of knowing the outcome is reduced is related to the probability of each possible outcome. In the extreme case where the outcome for a biased coin is always, say, heads ($P(H) = 1$) the information content for any given coin toss is reduced to zero bits. In general, for a number of possible outcomes v_i each with probability $P(v_i)$, the information content I of knowing the outcome is given by Equation 3 (Russell and Norvig 1995).

$$I(P(v_1), \ldots, P(v_n)) = \sum_{i=1}^{n} -P(v_i) log_2 P(v_i) \tag{3}$$

Information content can be used as a method for systematically selecting one attribute from a range of possible attributes to use in partitioning the set of objects T. For each possible partition of the set T with respect to a particular attribute l, the information gained by using that attribute can be calculated. This calculation involves estimating a set of probabilities associated with the partitioned sets T_i as a function of the ratio of objects in each partitioned set to the total number of objects (Russell and Norvig 1995). The attribute that results in the largest information gain should be the optimal attribute with which to partition the set T, since it provides more information about the decision tree than any other attribute.

3.1. *Induction example*

It is possible to provide an illustration of the induction algorithm in operation. The illustration is loosely based on experience with the KC data capture application, and concerns the accuracy of five ς-calculus objects each with just two categorical attributes, density and type, shown in Equation 4. Three different types of telecommunications point features are illustrated: 'pole' features are familiar telegraph poles used to support overhead cables; 'kiosk'

features are the familiar telephone kiosks, while 'cabinet' features are the street-level boxes used to house cable joints. For the purposes of this example, the objects have only one qualitative spatial attribute, the relative spatial density of features.

$$\begin{aligned} T \;=\; \{&o_1 = [\textit{density} = \text{“dense”}, \textit{type} = \text{“pole”}],\\ &o_2 = [\textit{density} = \text{“dense”}, \textit{type} = \text{“kiosk”}],\\ &o_3 = [\textit{density} = \text{“sparse”}, \textit{type} = \text{“cabinet”}],\\ &o_4 = [\textit{density} = \text{“sparse”}, \textit{type} = \text{“pole”}],\\ &o_5 = [\textit{density} = \text{“sparse”}, \textit{type} = \text{“kiosk”}]\} \end{aligned} \tag{4}$$

Where digital data is derived from paper maps, such as for the KC data capture project, high feature density may be associated with poor positional accuracy. Densely packed features on hardcopy maps are often deliberately displaced for cartographic reasons in addition to being harder to understand and digitise. The lower positional accuracy of such features often persists when digital data is derived from these cartographic products. In some cases, however, positional accuracy may be low regardless of feature density. In the case of the KC database, cabinet features on the original plant-on-plan maps explicitly use a symbology that obscures any precise location. Although the induction algorithm can have no 'understanding' of these sorts of processes, the induction algorithm is sensitive to data exhibiting these types of relationships. When shown a data set where low accuracy and high feature density are coincident it should be able to derive a rule or set of rules that embody this relationship.

Accordingly, the five objects in the set T have been categorised into two sets denoting low (C_l) and high (C_h) accuracy features, shown in Equations 5 and 6 respectively. The categories are broadly speaking as would be expected according to each object's spatial density attribute, with one object, o_3 a cabinet, exhibiting low accuracy C_l despite its low spatial density.

$$C_l \;=\; \{o_1, o_2, o_3\} \tag{5}$$

$$C_h \;=\; \{o_4, o_5\} \tag{6}$$

The induction process for this example is illustrated in Table 1, which expands on each step of the induction process. The result of this induction process is a simple decision tree, shown in Figure 1. The decision tree is automatically derived from the induction algorithm, but is a reflection of the more general processes behind the training set data. Having used induction to build a decision tree, it is possible to then categorise objects outside the original training set. For example, the object $o_6 \triangleq [\textit{density} = \text{“dense”}, \textit{type} = \text{“cabinet”}]$ was not part of the training set, but an examination of the decision

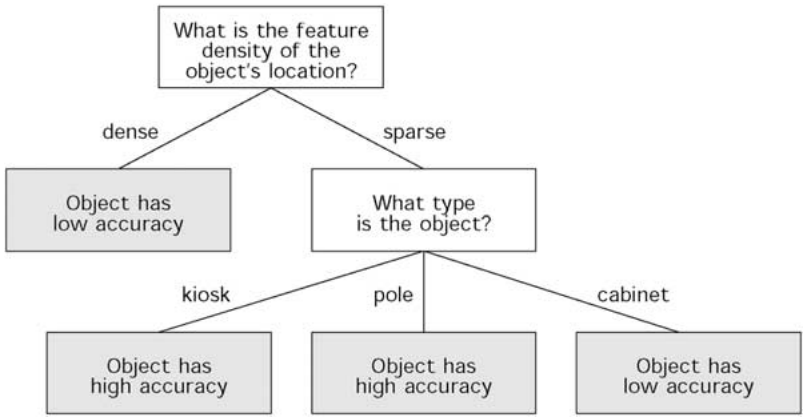

Figure 1. Example induction process results.

tree in Figure 1 reveals that such an object would be categorised as having low accuracy.

4. Optimising the induction algorithm

While naïve, the example in §3.1 above does illustrate how the core induction algorithm can operate for a very simple quality assessment. The inductive learning algorithm described above was used as the basis for an 'inductive data quality capture tool', described in §5. However, a number of optimisations were necessary before the induction algorithm could be incorporated into software intended for practical application. This section reviews the optimisations used in the software.

4.1. *Support for continuous attributes*

A common feature of all induction algorithms is that they are essentially categorical and operate only upon discrete information. While a categorical induction algorithm can be useful in many contexts, most data demands some quantitative capabilities. For example, imagine a training set that includes three polygonal objects with 'area' attribute values of 10.0 m^2, 10.1 m^2 and 100.0 m^2. The inductive learning algorithm would by default treat each area attribute value as a *separate* category. This is technically undesirable since treating continuous information as discrete quickly results in large fragmented decision trees riddled with decisions that yield minimal information gain. However, it is also semantically undesirable since we would probably intuitively expect 10.0 m^2 and 10.1 m^2 to appear in a different category to 100.0 m^2, but the same category as each other.

Table 1. Example induction process iterations

	Induction step	Details
0.1	Start induction process with T, C_l and C_h	$T = \{o_1, o_2, o_3, o_4, o_5\}$, $C_l = \{o_1, o_2, o_3\}$, $C_h = \{o_4, o_5\}$
1.1	Check for empty T	$T \neq \varnothing$
1.2	Check whether T contains objects of only one category	$T \not\subseteq C_l$ $T \not\subseteq C_h$
1.3	Partition T with first attribute, type.	$T_p = \{o_1, o_4\}$ $T_k = \{o_2, o_5\}$ $T_c = \{o_3\}$
1.4	Calculate information gain for type	$Gain(type) = I(\frac{3}{5}, \frac{2}{5}) - (\frac{2}{5}I(\frac{1}{2}, \frac{1}{2}) + \frac{2}{5}I(\frac{1}{2}, \frac{1}{2}) + \frac{1}{5}I(\frac{1}{1}, \frac{0}{1})) =$ 0.171 bits
1.5	Partition T with second attribute, density.	$T_d = \{o_1, o_2\}$ $T_s = \{o_3, o_4, o_5\}$
1.6	Calculate information gain for density	$Gain(density) = I(\frac{3}{5}, \frac{2}{5}) - (\frac{3}{5}I(\frac{2}{3}, \frac{1}{3}) + \frac{2}{5}I(\frac{2}{2}, \frac{0}{2})) = 0.420$ bits
1.7	Create new decision node using attribute with highest information gain and reiterate process.	Reiterate with T_d (2.1) and T_s (3.1)
2.1	Check for empty T_d	$T_d \neq \varnothing$
2.2	Check whether T_d contains objects of only one category	$T_d \subseteq C_l$ so iteration terminates with new leaf
3.1	Check for empty T_s	$T_s \neq \varnothing$
3.2	Check whether T_s contains objects of only one category	$T_s \not\subseteq C_l$ $T_s \not\subseteq C_h$
3.3	Partition T_s with first attribute, type.	$T_{s,p} = \{o_4\}$ $T_{s,k} = \{o_5\}$ $T_{s,c} = \{o_3\}$
3.4	Calculate information gain for type	$Gain(density, type) = I(\frac{2}{3}, \frac{1}{3}) - (\frac{1}{3}I(\frac{1}{1}, \frac{0}{1}) + \frac{1}{3}I(\frac{1}{1}, \frac{0}{1}) + \frac{1}{3}I(\frac{1}{1}, \frac{0}{1})) =$ 0.918 bits
3.5	Partition T_s with second attribute, density.	$T_{s,d} = \varnothing$ $T_{s,s} = \{o_3, o_4, o_5\}$
3.6	Calculate information gain for density	$Gain(density, density) = I(\frac{3}{3}, \frac{0}{3}) - (\frac{1}{3}I(\frac{1}{1}, \frac{0}{1}) + \frac{1}{3}I(\frac{2}{2}, \frac{0}{2})) = 0.000$ bits
3.7	Create new decision node using attribute with highest information gain and reiterate process.	Reiterate with $T_{s,p}$ (4.1), $T_{s,k}$ (5.1) and $T_{s,c}$ (6.1)
4.1	Check for empty $T_{s,p}$	$T_{s,p} \neq \varnothing$
4.2	Check whether $T_{s,p}$ contains objects of only one category	$T_{s,p} \subseteq C_h$ so iteration terminates with new leaf
5.1	Check for empty $T_{s,k}$	$T_{s,k} \neq \varnothing$
5.2	Check whether $T_{s,k}$ contains objects of only one category	$T_{s,k} \subseteq C_h$ so iteration terminates with new leaf
6.1	Check for empty $T_{s,c}$	$T_{s,j} \neq \varnothing$
6.2	Check whether $T_{s,c}$ contains objects of only one category	$T_{s,c} \subseteq C_l$ so iteration terminates with new leaf

The discretisation of continuous information is a common problem in learning systems (Susmaga 1997) and a wide range of discretisation algorithms have been proposed. The software described in the following section relies on a simple heuristic for discretisation, an approach also used by Walker and Moore (1988). The heuristic uses measures of spread to categorise the population of values for a particular continuous attribute into up to five separate categories. The approach is effective, simple and can operate unsupervised, but could easily be replaced by more sophisticated discretisation algorithms, such as cluster analysis.

4.2. *Spatial parameters*

A variety of spatial information naturally lends itself to a discrete representation, such as topological information. Spatially continuous information, such as coordinate location, must be discretised before induction in a similar way to continuous aspatial information, for example by using cluster analysis. Spatial information can be very rich and in addition to locational or topological information, other discretised derived spatial parameters can be included in the induction process. The inductive quality capture tool described in §5 automatically calculates a local measure of spatial density, geometric complexity and area or length where appropriate for each spatial object. The derived spatial parameters can then be discretised and utilised in the induction algorithm as described above.

It is worth noting that the choice of spatial parameters used introduces an element of circularity to the induction process. Feature density was already known to be important to data quality for the KC application (see §3.1), before it was included as an attribute to be used in the inductive learning algorithm. In general, the algorithm will only provide reasonable results if supplied with relevant information (see §4.4 below). Deciding what spatial (or aspatial) attributes are likely to be relevant may require some prior knowledge or experience of the problem domain. In practice, the task of deciding which attributes to include is not as difficult as might be imagined, since the inductive learning algorithm works best in information rich environments. As a general principle, ‘more is better’: the performance of the algorithm will most likely be improved by making any and all possible spatial and aspatial attributes available to the inductive learning algorithm.

4.3. *Majority classification*

There are two points within the induction process when arbitrary categorisations need to be used. The first point occurs when the training set for a particular iteration is empty, $T = \varnothing$ (see §3). $T = \varnothing$ occurs when a training

set has no objects that exhibit a particular value for an attribute being used to partition that training set. The second point, as suggested in §3, occurs when conflicting information exists and two objects with identical attributes belong to different categories. In reality both cases do occur, and the inductive quality capture tool uses a majority classification heuristic to provide a basis for an otherwise arbitrary categorisation. By looking at the range of different outcomes in the training set, or in the training set of the parent iteration in the case of $T = \varnothing$, the inductive quality capture tool assigns a new decision with the most populous outcome in that set. The assumption is that, on balance and in the absence of better information, the category with the majority of instances in the training set is the more likely outcome.

4.4. *Overfitting*

The inductive learning algorithm is far from infallible. A problem common to learning algorithms generally occurs when a learning algorithm infers meaningless patterns from a data set, termed *overfitting* (Russell and Norvig 1995). In particular, if the training set is unrepresentative or too small, the algorithm is much more likely to derive rules that relate to no particular processes or are entirely coincidental. In order to provide some guidance as to whether the training process has been successful, the induction algorithm reserves a portion of the training set, approximately one-third of the data, for cross-validation purposes. Having produced a decision tree using two-thirds of the training set, the decision tree is then used to deduce the correct categorisation for the remaining third of the training set. These results can be compared with the actual categorisations in the reserved third of the training set, to provide a guide as to how successful the training process was. Low classification accuracies indicate the training set is unrepresentative or too small and the training set needs to be extended.

There are two extreme cases that may cause the induction process to fail despite cross-validation. First, the cross-validation process may produce low classification accuracies for any training set drawn from a data set where spatial data quality is strongly related to some attribute which is not available in that data set. For example, when digitising map-based data, positional accuracy may be strongly dependent on the digitiser operator. If there exists no record of which operators digitised which features and there exists no discernible pattern in the spatial or thematic distribution of different digitiser operator's work (ie operators worked on features of every type and in every spatial region) then it is likely that the inductive learning algorithm would never achieve satisfactory cross-validation results. Second, high classification accuracies could be incorrectly indicated where spatial data quality is both strongly spatially autocorrelated and the training set happens to coincide with

a relatively homogeneous region or set of regions. This problem is more difficult to deal with, as it will result in artificially high classification accuracies that indicate training has been more successful than in fact has been the case. In practice, neither of these effects were observed, and both are relatively remote possibilities. For data sets that do exhibit these characteristics, the inductive learning algorithm is unsuitable and more conventional methods of data quality capture would be necessary.

4.5. *Spatial inference*

Following training, the decision tree should be able to make reasonable decisions regarding the quality of the geospatial objects from which the training set was drawn, even objects with attribute values that the decision tree has not encountered during training. Inevitably, the trained decision tree may come across objects that it cannot categorise because a completely new attribute value arises that was not present within the training set. Assuming the training set was representative, such situations should be infrequent. Rather than just abandon the automatic data quality assessment for these objects, the inductive quality capture tool attempts to match the problem object with a similar nearby object for which the attribute can be resolved. By further assuming the existence of spatial autocorrelation, it should be reasonable to substitute the nearest similar object for the problem object if the decision process stalls occasionally. The assumption of autocorrelation does not always hold. Many geographic features are not autocorrelated and in such cases the spatial inference mechanism should not be used. However, autocorrelation is undoubtedly an important factor in a wide range of geographic phenomena (Tobler 1970) and the assumption of autocorrelation will normally be a valid one. As an illustration of the spatial inference mechanism, in the example in §3.1 the trained decision tree in Figure 1 would have difficulty with an object o_7 = [*density* = "very dense", *type* = "cabinet"], since the attribute value "very dense" did not occur in the training set. In such a case, the spatial inference mechanism would be free to substitute the density attribute value of the nearest "cabinet" object. If such an object exists and the attribute is spatially autocorrelated, the object is more likely to have the density attribute value "dense" rather than "sparse" by virtue of being nearby and consequently ought to be a reasonable substitute.

4.6. *Parallel induction*

The automatic data quality capture technique described so far would be very useful for deriving decision trees which could be used to infer the quality of geospatial objects in terms of a *single* quality element. For example, the

algorithm could train a decision tree to infer accuracy *or* to infer lineage for a data set. However, it is very likely that for a given set of geospatial objects, accuracy, lineage and indeed any other quality element may vary independently of each other. Further, a particular quality element may have a number of attributes that also vary independently. The inductive quality assessment task can be viewed as a number of parallel induction tasks based on a training set categorised according to each attribute on each of the quality elements present in the training set.

This study developed a simple extension to the conventional induction algorithm outlined above, which is able to perform the induction process in parallel for several categorisations. In common with the conventional induction algorithm, the parallel induction algorithm uses a single training set and produces a single decision tree. At any given induction step the attribute used to partition the training set can be selected according to the total information gain produced by that attribute. Since information content is additive, the total information gain can be calculated from the summed information gain for each individual category family. Attributes can then be selected on the basis of maximal information gain across a range of categories. The result is that while a decision may be sub-optimal for an individual category at an individual iteration, overall the system still results in an efficient decision tree that should be able to resolve a number of categorisations at each step, effectively performing several categorisations at once.

For example, in the KC data capture process, certain feature types were captured by digitising from existing maps, while others were captured through field resurvey. In addition to the accuracy classifications C_l and C_h in §3.1, the lineage of geospatial data might be represented with two categories D_r, containing features that have been resurveyed, and D_d containing features have been digitised without resurvey, as in Equations 7 and 8.

$$D_r = \{o_2, o_3, o_5\} \tag{7}$$

$$D_d = \{o_1, o_4\} \tag{8}$$

The induction process can then proceed much as Table 1, independently calculating information gain for both accuracy and lineage, and calculating the optimal partition for all categories. Table 2 provides the first iteration of this revised parallel induction process, while Figure 2 provides a summarised version of the decision table resulting from the parallel induction process.

Table 2. Parallel induction process: first iteration

	Induction step	Details
0.1	Start induction process with T, C_l, C_h, D_r and D_d	$T = \{o_1, o_2, o_3, o_4, o_5\}$, $C_l = \{o_1, o_2, o_3\}$, $C_h = \{o_4, o_5\}$, $D_r = \{o_2, o_3, o_5\}$, $D_d = \{o_1, o_4\}$
1.1	Check for empty T	$T \neq \varnothing$
1.2	Check whether T contains objects of only one category	$T \not\subseteq C_l$ $T \not\subseteq C_h$ $T \not\subseteq D_r$ $T \not\subseteq D_d$
1.3	Partition T with first attribute, type.	$T_p = \{o_1, o_4\}$ $T_k = \{o_2, o_5\}$ $T_c = \{o_3\}$
1.4.1	Calculate information gain for type with accuracy	$Gain_C(type) = I(\frac{3}{5}, \frac{2}{5}) - (\frac{2}{5}I(\frac{1}{2}, \frac{1}{2}) + \frac{2}{5}I(\frac{1}{2}, \frac{1}{2}) + \frac{1}{5}I(\frac{1}{1}, \frac{0}{1})) =$ 0.171 bits
1.4.2	Calculate information gain for type with lineage	$Gain_D(type) = I(\frac{3}{5}, \frac{2}{5}) - (\frac{2}{5}I(\frac{2}{2}, \frac{0}{2}) + \frac{2}{5}I(\frac{2}{2}, \frac{0}{2}) + \frac{1}{5}I(\frac{1}{1}, \frac{0}{1})) =$ 0.971 bits
1.5	Partition T with second attribute, density.	$T_d = \{o_1, o_2\}$ $T_s = \{o_3, o_4, o_5\}$
1.6.1	Calculate information gain for density with accuracy	$Gain_C(density) = I(\frac{3}{5}, \frac{2}{5}) - (\frac{3}{5}I(\frac{2}{3}, \frac{1}{3}) + \frac{2}{5}I(\frac{2}{2}, \frac{0}{2})) = 0.420$ bits
1.6.2	Calculate information gain for density with lineage	$Gain_D(density) = I(\frac{3}{5}, \frac{2}{5}) - (\frac{3}{5}I(\frac{2}{3}, \frac{1}{3}) + \frac{2}{5}I(\frac{1}{2}, \frac{1}{2})) = 0.020$ bits
1.7	Create new decision node using attribute with highest total information gain and reiterate process.	$Gain(type) = 0.171 + 0.971 = 1.142$bits $Gain(density) = 0.420 + 0.020 = 0.440$bits so reiterate with T_p, T_k and T_c
		...

5. Implementation results

An inductive quality capture tool was implemented using the inductive learning algorithm outlined above. The tool offers a simple interface to help GIS users to incorporate quality information into their data set during data capture. The quality capture tool is intended to work alongside conventional geospatial data capture streams. In particular, it is aimed at legacy data capture projects, such as that undertaken by KC.

5.1. *Choosing the training set*

Use of the inductive quality capture tool begins with a pilot assessment of the quality of a small area of the legacy paper map data being captured. This pilot quality assessment forms the training set for the inductive learning algorithm. In the case of KC, it proved entirely feasible to derive a picture of the history and accuracy of the KC data without the need for resurvey. Simply by looking through the project documentation, familiarity with the source maps and by

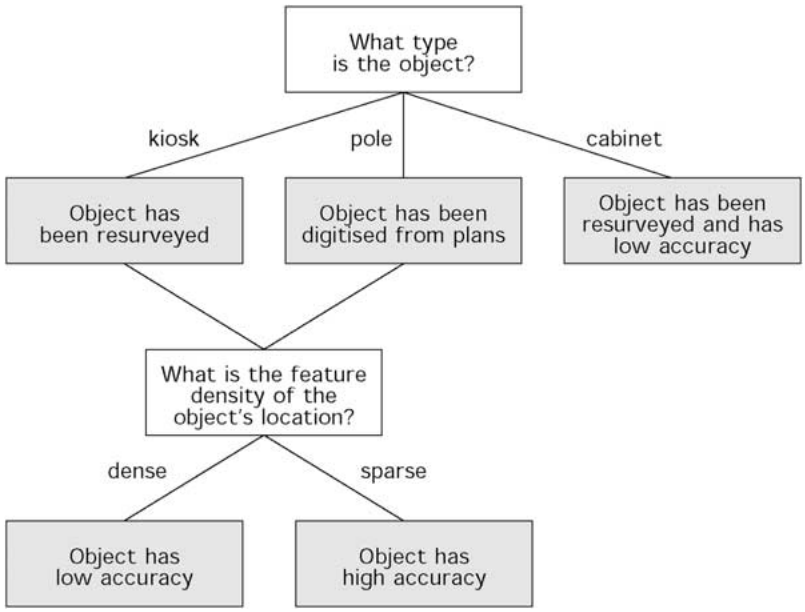

Figure 2. Example parallel induction process results.

talking with the KC, SDS and Informed Solutions employees it was possible to produce a credible pilot quality assessment. Perversely, a significant body of quality information associated with legacy paper maps will usually be lost during the migration to digital mapping. Lineage information on the provenance of maps and map features is well known to engineers used to handling those maps. Levels of accuracy, precision and detail are often implied by the physical limitations of the map, limitations which do not apply once the map is digitised. In other cases it might be necessary to embark upon relatively expensive resurvey. In the case of legacy data capture the value of such informally developed quality information should never be discounted. The pilot assessment was conducted along conventional lines, based on the procedures set out in the Spatial Data Transfer Standard (SDTS, US Geological Survey 1999). The detailed choice of quality elements used is external to the induction algorithm, and any quality elements or standard could be used. Initial experiments with other common standards, namely Spatial Archive and Interchange Format (SAIF, Geographic Data BC 1996) and the European draft standard (CEN/TC287 1996) proved just as successful as using SDTS as a basis.

It is worth noting that the pilot quality assessment used for the training set need not be located within a single contiguous geographic area. For the purposes of the core induction algorithm, the pilot quality assessment

can operate using a training set composed of features that are geographically dispersed across the study area. However, two practical considerations militate against using such dispersed training sets. First, it will usually be much more efficient from the point of view of data capture to perform the pilot quality on a single contiguous sub-set of the study area rather than perform a piecemeal assessment over the entire study area. Second, the induction optimisation routines may assume that the training set is not spatially dispersed. In particular, the calculation of feature density mentioned in §4.2 assumes that the spacing of features in the training set is characteristic of the study area generally. If the training set is spatially dispersed this assumption will not hold.

5.2. *Tool architecture*

The inductive quality capture tool was programmed using Java object-oriented programming language (OOPL). The tool communicates with a geospatial database via a three-tier client-server architecture built using Java remote method invocation (RMI), related to the common object request broker architecture (CORBA). Potentially, this architecture means the tool could be used to interface with almost any database. In the case of this study, the core spatial database functionality was provided by Laser-Scan Gothic OO GIS. The Laser-Scan database was modified to allow objects in the spatial database to be associated with quality objects, as described in Duckham and Drummond (1999). Using RMI, the relationships between quality and other database objects can be controlled transparently by any Java program, such as the inductive quality capture tool.

The tool interface, shown in Figure 3 acts as a data import filter, allowing the pilot data set to be imported and quality assessment information added to this pilot data set. The pilot data set for this study was drawn from the KC data supplied by SDS in the form of CLIFF files, the intermediate text file format used by the KC project for data transfer. The tool as depicted in Figure 3 has four linked windows. The main map window, in the top left of Figure 3, shows a portion of the pilot data set and the menus needed to control tool operation. Once loaded, the pilot data set is annotated with quality information gathered, in this case, from the informal sources described in §5.1. In order to annotate the pilot data set with quality information a further three different types of window are needed, shown in Figure 3. Clockwise from the main map window, a geospatial object selection window displays the attributes of geospatial objects selected from the main map window, while a selected quality object and a quality attribute window allow individual quality objects to be defined and associated with selected geospatial objects.

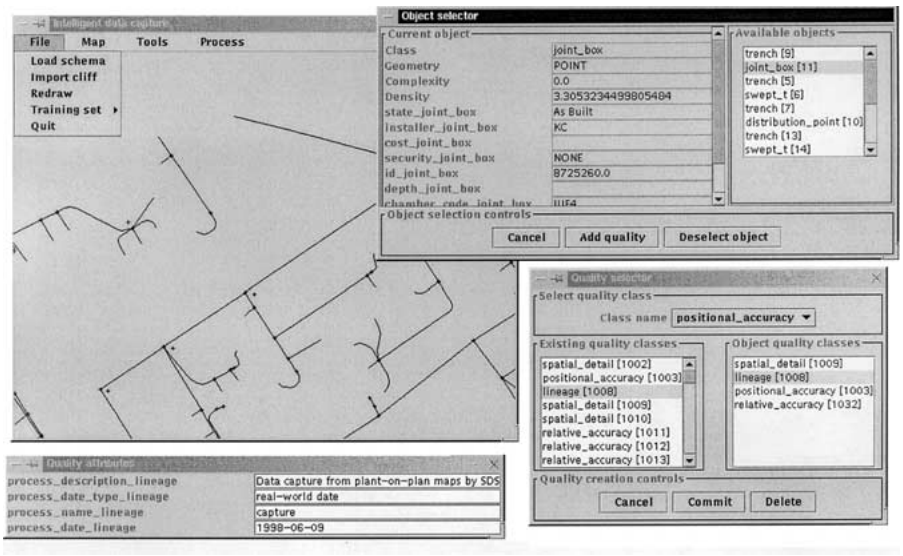

Figure 3. Pilot quality assessment

Once the quality assessment information has been added to the pilot data, this information can be used as the basis of a wider quality assessment. The tool uses the pilot data set as a training set for the inductive learning algorithm. The geospatial data in the training set is categorised into a number of separate category families according to its associated quality objects' attributes. Using this training set the quality capture inductive learning algorithm looks for patterns in the geospatial data that imply patterns in the quality data. The product of the induction algorithm is a decision tree tailored to the particular features of the telecommunications data being captured. Once created, this decision tree can be applied to the remaining data being captured, automatically deducing quality information. Java RMI is used to manage the flow of information between the inductive quality capture tool and the Laser-Scan database, although the architecture ensures that this information flow is completely hidden from the tool user.

5.3. *Tool performance*

The induction algorithm is guaranteed to infer a decision tree from a representative training set of geospatial objects and their associated quality. This decision tree can then be used to derive quality information for the full data set from which the training set is derived. Based on the KC data capture project, the technique was tested on a small section total area, approximately half of a 1 km^2 area located at UK National Grid coordinates (510000,

434000). The discussion in §4.4 highlighted the problems with overfitting where unrepresentative or small data sets infer meaningless patterns. In order to increase the likelihood of meaningful results, following training and cross-validation the tool interface displays a dialogue box that reports the classification accuracy of the training process, along with some guidance as to what that accuracy means and whether the pilot data set should be extended. As a rule of thumb, this study suggested that the best results were produced by pilot assessments covering between 5 and 10% of the total number of features. Such assessments generally resulted in classification accuracies of 80% or greater. Assessments of less than 5% of the total number of features were much more likely to be unpredictable or unreasonable. As indicated in §4.4, data exhibiting certain unusual characteristics may produce consistently low classification accuracies or otherwise poor performance. While no such problems were encountered in this study, users should be aware that inductive learning may not be suitable for certain special data sets.

As a last line of defence, all automatically generated quality objects are themselves associated with a quality object (a *meta-quality* object) that reports both the fact that the quality object was automatically generated and a simple justification of the inductive process leading to the decision to use that quality object. Where majority classification (see §4.3) has been used for example, the meta-quality object reports this fact and the size of the majority. Intuitively this meta-quality information represents the degree of uncertainty associated with a particular piece of quality information. The meta-quality information provides the basis of 'quality audit', so that following the quality capture process the original data sets can still be retrieved, and automatically derived quality information can always be distinguished from manually derived quality information.

6. Further work

The work so far has indicated that inductive learning is a suitable technique for efficient capture of scarce spatial data quality information. However, there remains a question mark over how the results of such an inductive quality capture exercise should be interpreted. Even assuming the training set is representative of the full data set, it is a moot point as to what extent the quality information produced by the induction algorithm can be considered 'correct'. There is a dearth of research addressing the reliability of quality assessments, and it is difficult to see how the reliability of quality information could be tested using conventional experimental methods. Conceivably, a comparison between repeated independent quality assessments would yield an idea of how accurate a particular quality assessment procedure is. Such

experiments have not been performed and, given the difficultly in encouraging companies to perform a single quality assessment, it is implausible to expect the same companies to perform a statistically representative set of quality assessments in order to derive meta-quality information about the reliability of their quality assessment procedure. In the absence of such a mechanism for externally verifying the reliability of spatial data quality information, it is difficult to make any sweeping judgments about the inductive quality capture tool's performance, other than to say that the results appear to be reasonable. Clearly, further research into the semantics and veracity of spatial data quality information would be beneficial, both to better evaluate the results of this research and more widely for research into spatial data quality.

Other work is also suggested by this research. Cross-validation is an effective, but relatively crude mechanism for ensuring the inductive learning algorithm is operating efficiently. Presenting a single statistic, classification accuracy, may hide important information about the detailed characteristics of the induction process. Further work exploring alternative techniques for providing a more sophisticated impression of the spatial and thematic characteristics of data quality would be useful. Visualisation of the spatial and thematic distribution of the multi-dimensional data quality information, for example, might help users gain some insight into the nature and extent of likely spatial data quality issues during training. Similarly, sensitivity analysis of the training process applied to different training sets and attributes would help provide the user with a more complete picture of the key factors influencing the efficacy of the inductive learning algorithm.

7. Conclusions

The failure to collect quality information is a self-perpetuating reason for a widespread failure to incorporate quality management procedures in digital data capture projects. The already high cost of collecting geospatial data, coupled with the high levels of competition in industries like telecommunications, mean that such industries are unlikely to embrace quality management of geospatial data on short-term financial grounds alone. The value of the inductive quality capture tool within the error-aware GIS architecture is that it maximises the efficiency of quality assessment, requiring only a small fraction of the information produced during full quality assessment to operate. Potentially, introducing low-cost quality capture techniques is the first step in breaking the cycle that prevents companies collecting and using data quality information for geospatial data sets.

The induction algorithm at the heart of the inductive quality capture tool uses a pilot data set to infer general rules relating quality to geospatial objects. Individual geospatial objects in the pilot data set are categorised according to their quality. The induction algorithm is able to build a decision tree based on the spatial and aspatial characteristics of the geospatial objects, while performing a cross-validation on a reserved portion of the pilot data set. Assuming the cross-validation indicates the training process has been successful, this decision tree can be used to automatically infer quality more generally across the geospatial data set. Experiences during this study support the view that induction when applied to automated quality capture can produce reasonable results, while automatically generated meta-quality information can provide guidance as to the verity of inferred quality information.

Acknowledgements

The cooperation and assistance of the staff of Survey and Development Services (SDS), Informed Solutions and Kingston Communications (KC) was essential to the research. In particular, the authors would like to thank John McCreadie and Elspeth Rodger of (SDS), Justin Hassal (Informed Solutions) and Paul Rickatson (KC). Gothic Software was kindly supplied on a development license from Laser-Scan, UK. The research was supported by NERC Award No. GT19/96/9/EO under a CASE agreement with SDS. Matt Duckham is currently supported at Keele University by an EPSRC project, grant number GR/M 56685. Finally, the helpful and constructive comments of two anonymous reviewers are gratefully acknowledged.

References

Abadi, M. and Cardelli, L. (1996). *A Theory of Objects*. New York: Springer-Verlag.

Agumya, A. and Hunter, G. (1997). Determining Fitness for Use of Geographic Information, *ITC Journal* (2): 109–113.

Aspinall, R. (1992). An Inductive Modelling Procedure Based on Bayes Theorem for Analysis of pattern in Spatial Data, *International Journal of Geographical Information Systems* 6(2): 105–121.

Bastin, L., Wood, J. and Fisher, P. (1999). Visualisation of Fuzzy Spatial Information in Spatial Decision Making. In K. Lowell and A. Jaton (eds.), *Spatial Accuracy Assessment: Land Information Uncertainty in Natural Resources*, Chapter 18 (pp. 147–150). Michigan: Ann Arbor.

Bennet, D. and Armstrong, M. (1996). An Inductive Based Approach to Terrain Feature Extraction, *Cartography and Geographic Information Systems* 23(1): 3–19.

Campbell, G., Carker, L. and Egesborg, P. (1994). A GIS-Based Multipurpose Digital Cadastre for Canada Lands. In *FIG Congress XX*.

CEN/TC287 (1996). Draft European Standard: Geographic Information – Quality. Technical Report prEN 287008, European Committee for Standardisation.

Codd, E. (1970). A Relational Model of Data for Large Stored Data Banks. *Communications of the ACM* 13(6): 377–387.

Duckham, M. (2001). Object Calculus and the Object-Oriented Analysis and Design of an Error-Sensitive GIS, *GeoInformatica* 5(3): 261–289.

Duckham, M. and Drummond, J. (1999). Implementing and Object-Oriented Approach to Data Quality. In B. Gittings (ed.), *Integrating Information Infrastructures with GI Technology* (pp. 53–64). London: Taylor and Francis.

Egenhofer, M. and Frank, A. (1989). Object-Oriented Modeling: Inheritance and Propagation. In *Proceedings Auto-Carto 9* (pp. 588–598).

Geographic Data BC (1996). Spatial Archive and Interchange Format Release 3.2 Formal Definition. http://www.env.gov.bc.ca/gdbc/saif32/. Last modified 14 September 1999, last accessed 21 December 1999.

Goodchild, M. (1999). Measurement-Based GIS. In W. Shi, M. Goodchild and P. Fisher (eds.), *Proceedings of the International Symposium on Spatial Data Quality* (pp. 1–9).

Hankin, C. (1994). *Lambda Calculi: A Guide for Computer Scientists*. Oxford: Clarendon Press.

Heuvelink, G. (1998). *Error Propagation in Environmental Modelling with GIS*, Research monographs in GIS. London: Taylor and Francis.

Hunter, G. (1999). Reporting Spatial Data Quality: From Concepts to Reality. In W. Shi, M. Goodchild and P. Fisher (eds.), *Proceedings of the International Symposium on Spatial Data Quality* (pp. 343–353).

Kösters, K., Pagel, B.-U. and Six, H.-W. (1997). GIS-Application Development with GEOOOA, *International Journal of Geographical Information Systems* 11(4): 307–335.

Lanter, D. (1991). Design of a Lineage-Based Meta-Data Base for GIS, *Cartography and Geographic Information Systems* 18(4): 255–261.

Lanter, D. and Veregin, H. (1992). A Research Paradigm for Propagating Error in Layer-Based GIS, *Photogrammetric Engineering and Remote Sensing* 58(6): 825–833.

Mikhail, E. (1978). *Observations and Least Squares*. New York: IEP Dun Donnely.

Openshaw, S., Charlton, M. and Carver, S. (1991). Error Propagation: A Monte Carlo Simulation. In I. Masser and M. Blakemore (eds.), *Handling Geographical Information* (pp. 78–101). New York: Longman.

Qiu, J. and Hunter, G. (1999). Managing Data Quality Information. In W. Shi, M. Goodchild and P. Fisher (eds.), *Proceedings of the International Symposium on Spatial Data Quality* (pp. 384–395).

Quinlan, J. (1979). Discovering Rules by Induction from Large Collections of Examples. In D. Michie (ed.), *Expert Systems in the Micro-Electronic Age* (pp. 168–201). Edinburgh University Press.

Quinlan, J. (1983). Learning Efficient Classification Procedures and Their Application to Chess end Games. In R. Michalski, J. Carbonell and T. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Chapter 15 (pp. 463–482). California: Morgan Kauffmann.

Ramlal, B. and Drummond, J. (1992). A GIS Uncertainty Subsystem. In *Archives ISPRS Congress XVII*, Vol. 29.B3 (pp. 356–362).

Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice Hall.

Shannon, C. (1948). A Mathematical Theory of Communication, *The Bell System Technical Journal* 27: 379–423, 623–656.

Smith, B. (1996). NLIS in 1996: The Pilot Project Expands. *Mapping Awareness* 10(2): 22–24.

Susmaga, R. (1997). Analyzing Discretizations of Continuous Attributes Given a Monotonic Discrimination Function. *Intelligent data analysis* 1(3).

Tobler, W. (1970). A Computer Movie: Simulation of Population Change in the Detriot Region, *Economic Geography* 46: 234–240.

Unwin, D. (1995). Geographical Information Systems and the Problem of Error and Uncertainty. *Progress in Human Geography* 19(4): 549–558.

US Geological Survey (1999). SDTS standard. http://mcmcweb.er.usgs.gov/sdts/. Last modified 16 June 1999, last accessed 21 December 1999.

van der Wel, F., Hootsmans, R. and Ormeling, F. (1994). Visualization of Data Quality. In A. MacEachren and D. Taylor (eds.), *Visualization in Modern Cartography* (pp. 313–331).

van Elzakker, C., Ramlal, B. and Drummond, J. (1992). The Visualisation of GIS Generated Information Quality. In *Archives ISPRS Congress XVII*, Vol. 29.B4 (pp. 608–615).

Veregin, H. (1989). Error Modeling for the Map Overlay Operation. In M. Goodchild and S. Gopal (eds.), *Accuracy of Spatial Databases* (pp. 3–18). London: Taylor and Francis.

Walker, P. and Moore, D. (1988). SIMPLE: An Inductive Modelling and Mapping Tool for Spatially-Oriented Data. *International Journal of Geographical Information Systems* 2(4): 347–363.

Wesseling, C. and Heuvelink, G. (1993). Manipulating Qualitative Attribute Accuracy in Vector GIS. In: *Proceedings Fourth European Conference on Geographical Information Systems*, Vol. 1 (pp. 675–684).

Worboys, M., Hearnshaw, H. and Maguire, D. (1990). Object-Oriented Data Modelling for Spatial Databases. *International Journal of Geographical Information Systems* 4(4): 369–383.