# An algebraic approach to automated geospatial information fusion

Matt Duckham[*], Mike Worboys[†]

[*]Department of Geomatics
University of Melbourne
Victoria 3010, Australia
mduckham@unimelb.edu.au

[†]NCGIA, Department of Spatial Information Science and Engineering
University of Maine
Orono, ME 04469, USA
worboys@spatial.maine.edu

**Abstract**

This paper presents a new technique for information fusion. Unlike most previous work on information fusion, this paper explores the use of instance-level (extensional) information within the fusion process. This paper proposes an algorithm that can be used automatically to infer the schema-level structure necessary for information fusion from instance-level information. The approach is illustrated using the example of geospatial land cover data. The method is then extended to operate under uncertainty, such as in cases where the data is inaccurate or imprecise. The paper describes the implementation of the fusion method within a software prototype. Finally, the paper discusses several key topics for future research, including applications of this work to spatial data mining and the semantic web.

**Keywords** interoperability, qualitative spatial reasoning, geospatial ontology, uncertainty, knowledge representation

## 1 Introduction

*Information fusion* is the process of integrating information from diverse sources to produce new information with added value, reliability, or utility (cf. Wald, 1999). Information fusion is a basic function of any information system. The key problem in achieving information fusion is matching the diverse semantics of categories and relationships in the different information sources. This paper presents a new approach to fusion that is able to exploit knowledge of such semantics derived from *instances* in the data, rather than relying solely on the definitions of categories for matching semantics.

To illustrate, if all instances of "swamp fen" in database $A$ are classified as "inland marsh" in database $B$, we might infer that the category "inland marsh" subsumes the category "swamp fen." This inference does not depend on the semantics of "inland marsh" or “swamp fen,” and so might be performed by a person with no knowledge or experience of marshes and fens, or even (as in this paper) by an automated reasoning system. Although the example is grossly simplified, more sophisticated instance-based analysis can be achieved if the instances have more structure. Of particular relevance to this paper is the structure imposed by geospatial location (i.e. we have information about where the swamp fens and inland marshes are located).

In this paper we set the scene in section 2 with an overview of the approach taken in this paper and relevant literature. Section 3 presents the model and algebraic basis for instance-based information fusion, followed by a worked example in section 4. The process of reasoning from specific cases to general rules (as in the example above) is termed *inductive inference*. Inductive inference can be unreliable, particularly where instance-level data is uncertain (for example, if some instances are incorrectly classified as “swamp fens” or “inland marshes”). Consequently, allowing for uncertainty is a consideration in the instance-based fusion process, addressed in section 5. Section 6 contains a discussion of key implementation issues and description of a fusion system software prototype. Finally, section 7 concludes the paper with a summary and road map for future work.

## 2 Background

The semantics of an information source may be described using an ontology (defined as “an explicit specification of a conceptualization,” Gruber, 1993). The task of fusing information compiled using different ontologies is a classical problem in information science (e.g., Wiederhold, 1992). This problem continues to be an important research issue within many topics, including schema integration in databases (Kim and Sea, 1992; Lakshmanan et al., 1993; Sheth and Kashyap, 1993); semantic heterogeneity in interoperability (Vckovski, 1998; Sheth, 1999); schema matching in the semantic web (Berners-Lee et al., 2001; Doan et al., 2002); mediators (Wiederhold, 1992; Ullman, 2000; Tzitzikas et al., 2001); ontology-based information integration (Ganter and Wille, 1999; Guarino, 1998; Guarino et al., 1999; Fonseca et al., 2002); knowledge representation (Calvanese et al., 1998); and data warehousing (Widom, 1995; Zhou et al., 1995).

The overwhelming majority of existing techniques for information fusion are not automated, and focus solely on the schema itself (see Wache et al., 2001, for a review of ontology-based information fusion). Automation is hard to achieve because the semantics of information are difficult or impossible to precisely define. Some techniques have been developed for automating ontology integration based on natural language processing of schemas or schema descriptions, or through analysis of the structure of the schema (see Rahm and Bernstein, 2001, for a survey of these techniques). However, these techniques are typically only applicable to relatively narrow application domains, and are unable to account for the mismatch between how information is defined and structured in an ontology and how information is actually used in practice. As a result, most studies of the application of ontology integration techniques to practical domains, such as geographic information (e.g. Kokla and Kavouras, 2001; Kavouras and Kokla, 2002; Worboys and Duckham, 2002), are at best partially automated and rely on high levels of human application domain expertise to define the mapping between the ontologies of different information sources.

Conventional information fusion techniques, including work on ontology-driven information systems (e.g., Guarino, 1998; Fonseca and Egenhofer, 1999), are concerned with schema-level (or *intensional*) information about the definition and structure of an information source. Focusing solely on schema-level information ignores a valuable source of instance-level (or *extensional*) information about how concepts defined in the schema are actually used in an information source (Rahm and Bernstein, 2001; Berlin and Motro, 2001). A handful of techniques have been developed that are able to exploit such instance-level information. Doan and collaborators have explored using machine learning and probabilistic models to identify the most likely mappings between elements in the schemas of the information sources to be integrated (Doan et al., 2000, 2002). Li and Clifton have developed a technique that first classifies patterns of structure and values within instance-level information and then uses the classifier algorithm to train a neural network to identify similar elements within an information source schema (Li and Clifton, 1994, 2000). In this paper we present a new technique for automate the fusion process using instance-level information that is based on an algebraic rather than a probabilistic approach.

## 2.1 Overall approach

We can summarize our approach and contrast it with other work on information fusion using Figures 1 and 2. Most previous work on information fusion is concerned primarily with integrating the schema-level definitions of information, as shown in Figure 1. This process typically requires the intervention of a human domain expert to identify the shared ontological structure between the different schemas. For example, in order to fuse two different land cover maps of a particular region, domain expertise would be needed to identify the schema-level relationships between the ontologies for those two maps. The fusion process would then proceed based on this domain expert's knowledge, such as the knowledge that the class "Forest" in one map is the same as "Woodland" in another map.

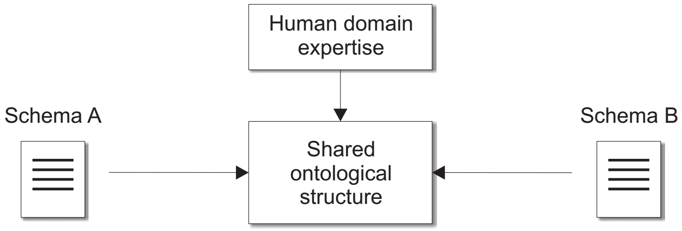


Figure 1: Summary schema-level information fusion

By contrast, the approach taken in this paper is to use both schema-level definitions and instance-level examples within the information fusion process, shown by Figure 2, which includes all of Figure 1. In order to operate, the fusion process does not require any schema-level information, other than the conceptual schemas (data models) for the data sets to be fused (as provided, for example, by a database). Automated pattern recognition techniques can be used to identify shared structure within the instance-level examples, such as shared lexical, geometric, or topological structure. This shared instance-level structure may then be used to infer shared schema-level structure. Continuing the land cover example above, if we happen to notice that all the “Forest” regions in one map are “Woodland” regions in the other map (instance-level structure) we might infer that the land cover class “Forest” is subsumed by the class “Woodland” (schema-level structure).

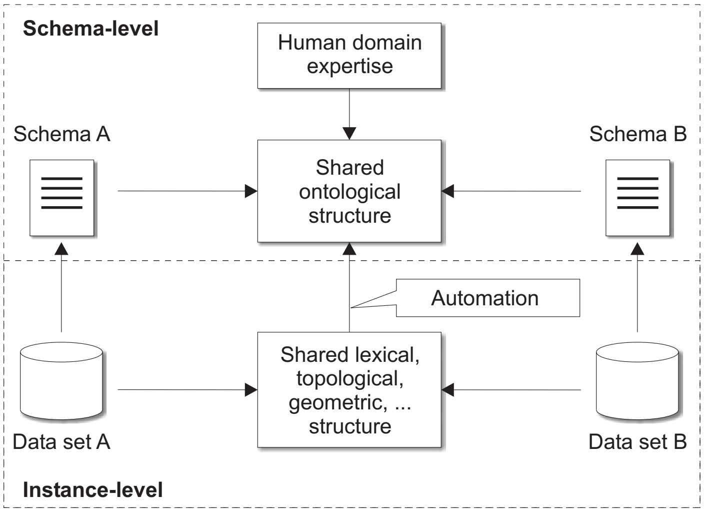


Figure 2: Summary of schema- and instance-level information fusion

Where available, additional intensional information can be incorporated into the fusion process (see sections 3.8 and 4.1). This additional information might come directly from human domain experts. However, considerable research effort over recent years has been directed at the problem of formalizing, storing, and processing the ontologies and contexts associated with conceptual schemas, especially within the domain of geographic information (e.g., Brodaric and Gahegan, 2002; Fonseca et al., 2003). Such intensional information, perhaps stored within an *ontology-based information system* Guarino (1998), could potentially be incorporated into the fusion process using the same mechanism as for human domain expertise.

Note that potentially any instance-level structure may be used to drive the fusion process in Figure 2. The common geographic coordinate systems of geospatial information provides an ideal structure. For this reason, and because information fusion is a high-profile problem for geographic information science, most of the examples in this paper concern geospatial information. However, the approach is not limited to fusion of geospatial information, and the final section indicates some other non-spatial application areas are being explored.

In summary, the two main advantages of using instance-level information in addition to schema-level information in the fusion process are:

1. How concepts are defined is not necessarily the same as how they are used. Only by looking at instance-level information can we determine how concepts are actually used.

2. Instance-level information forms a rich source of examples that can be used to automate part or all of the fusion process. Understanding the semantics of schema-level information requires human expertise in all but the narrowest problem domains.

# 3 Fusion construction

This section provides an algebraic specification of our fusion algorithm, based in part on Worboys and Duckham (2002). The earlier paper presents an algebra of information fusion where schema-level information is related through the definition of a set of "atoms," which define the indivisible classification elements used to relate two schemas. The approach presented below represents a significant step forward from this earlier work, because the construction presented here requires no atoms. The process of atom definition in Worboys and Duckham (2002) requires domain human expertise, and so developing an atomless model is a prerequisite for automating the fusion process.

At certain points, the following material uses some standard results from algebra and lattice theory. Readers who are not familiar with these topics may find it helpful to refer also to section 4, where we provide a concrete worked example of the fusion construction.

## 3.1 Classifications

The first step in the fusion construction is to formally define a classification of a region. A region is assumed to be partitioned into a finite number of blocks. A classification is then an assignment of a value in a classification structure to each block of the partition. In this paper, we make the assumption that the classification structure is a finite join semilattice, so that each pair of classes has unique minimum class that subsumes them both. This assumption implies that each classification structure has a top element $\top$. A join semilattice is necessary in order to obtain the properties of the fusion construction, discussed below in section 3.7. Sometimes, it is convenient to add or include a bottom element $\perp$ to the semilattice. There is an elementary result in lattice theory that every finite join semilattice with $\perp$ is a lattice.

So, let $X$ be a region, and $\mathsf{part}(X)$ be the set of all partitions of the region. A *classification* of $X$ is a tuple

$$\mathcal{C} = \langle X, P, T, g \rangle$$

where function $g : P \rightarrow T$, $P \in \mathsf{part}(X)$, and $T$ is a join semilattice, with top element $\top$. Unless otherwise stated, we assume below that $X$ is fixed. In that case, we may unambiguously express a classification as

$$\mathcal{C} = \langle P, T, g \rangle$$

## 3.2 Morphisms between classifications

Let $\mathcal{C}_1 = \langle P_1, T_1, g_2 \rangle$ and $\mathcal{C}_2 = \langle P_2, T_2, g_2 \rangle$ be two classifications. Suppose there are functions, $\pi : P_1 \rightarrow P_2$ and $\tau : T_2 \rightarrow T_1$, such that

1. $\tau$ preserves semilattice joins.

2. For all $p_1 \in P_1, \tau g_2 \pi(p_1) \geq g_1(p_1)$

If these conditions are satisfied, we say that there is a *morphism* $\mu = \langle \pi, \tau \rangle : \mathcal{C}_1 \rightarrow \mathcal{C}_2$. The arrow diagram below shows the morphism configuration between two classifications.

$$\begin{tikzcd}
P_1 \arrow[r, "g_1"] \arrow[d, "\pi"'] \arrow[dr, phantom, "\tau g_2 \pi \geq g_1"] & T_1 \\
P_2 \arrow[r, "g_2"] & T_2 \arrow[u, "\tau"']
\end{tikzcd}$$

Two classifications $\mathcal{C}_1$ and $\mathcal{C}_2$ are *isomorphic* if there exists a morphism $\mu = \langle \pi, \tau \rangle : \mathcal{C}_1 \rightarrow \mathcal{C}_2$ such that inverse functions $\pi^{-1}$ and $\tau^{-1}$ exist with the property that $\mu^{-1} = \langle \pi^{-1}, \tau^{-1} \rangle : \mathcal{C}_2 \rightarrow \mathcal{C}_1$ is also a morphism. We write $\mathcal{C}_1 \cong \mathcal{C}_2$. Note that in this case, $g_1 \subseteq \tau g_2 \pi \subseteq \tau^{-1} \tau g_2 \pi \pi^{-1} = g_1$, and so the full commutativity condition, $\tau g_2 \pi = g_1$ holds in this case.

### 3.3 Extensional forms of classifications

At the heart of the formal fusion construction is the observation that classifications can be built out of the blocks of their partitions. Informally, we can describe any class in a classification as the set of all instances of that class, i.e., a class can be defined with reference to its extension. To illustrate, one way to describe exactly what we mean by the class "elephant" is to refer to the set of all objects that we call elephants. This set would include all objects that were also instances of sub-classes of "elephant," for example, instances of the classes "Indian elephant" and "African elephant."

Realizing the importance of these extensional forms of classifications was one of the most frustrating and time-consuming stages in the development of our fusion construction. However, using the extensional forms of classifications results in a fusion construction that is, overall, much simpler. The extensional form of a classification makes explicit the link between the intensional and extensional information, and provides a formal definition of a class within the *context* of a particular data set. The formal construction of the extensional form of a classification is given below. It might be helpful to read the remainder of section 3 in parallel with section 4, where a concrete example is worked through.

Formally, let classification $\mathcal{C} = \langle P, T, g \rangle$ be given. Define $T' = \langle \mathrm{Im}(g) \rangle$ to be the join sub-semilattice of $T$ generated by $\mathrm{Im}(g)$, where $\mathrm{Im}(g)$ denotes the image of function $g$. Let function $g'$ be the restriction of $g$ to codomain $T'$. Then, there is an obvious morphism from $\mathcal{C} = \langle P, T, g \rangle$ to $\mathcal{C}' = \langle P, T', g' \rangle$.

We are now ready to construct the extensional form of the classification. The key observation is that each $x \in T'$ may be represented uniquely as the join of all elements of $\mathrm{Im}(g)$ less than or equal to $x$. Define the *extension function*, $e$, as:

$$e : T' \rightarrow \mathcal{P}(X)$$

$$e : x \mapsto \bigvee_{y \in \mathrm{Im}(g), y \leq t} g^{-1}(y) \text{ where } x = \bigvee_{y \in \mathrm{Im}(g), y \leq t} y$$

Now define $\overline{\mathcal{C}} = \langle P, \overline{T}, \overline{g} \rangle$, where $\overline{g} = eg'$ and $\overline{T} = \mathrm{Im}(e)$. By the uniqueness of the join construction, $e : T' \to \overline{T}$ is an injection, and therefore a bijection. It is possible to check that $e$ preserves joins. So $\overline{\mathcal{C}} \cong \mathcal{C}'$, and there is a morphism from $\mathcal{C}$ to $\overline{\mathcal{C}}$. The classification $\overline{\mathcal{C}}$ is called the extensional form of classification $\mathcal{C}$.

### 3.4 Regular classifications

The extensional form of a classification constructed above in a sense gives the classification pared down to its essential form. Elements in the semilattice that are not used or distinguished in the classification are eliminated or merged in the extensional form. In some cases, we wish to work with a classification that is already rid of its inessential elements. This is expressed by the following definition. A classification is called *regular* if it is isomorphic to its extensional form, that is, $\mathcal{C} = \overline{\mathcal{C}}$.

### 3.5 Refinement classifications

A partition $P'$ is a *refinement* of partition $P$ whenever the following condition holds.

$$\forall p' \in P' \; \exists p \in P \; p' \subseteq p$$

We may note that such a $p$ is unique.

Given a refinement $P'$ of $P$ and a classification $\mathcal{C} = \langle P, T, g \rangle$, we can construct a classification $\mathcal{C}' = \langle P', T, g' \rangle$, as shown in the following arrow diagram.

$$\begin{tikzcd}
P' \arrow[r, "g'"] \arrow[d, "\pi"'] & T \\
P \arrow[r, "g"] & T \arrow[u, "\mathrm{id}"']
\end{tikzcd}$$

Function $\pi$ assigns to each $p' \in P'$ the unique $p \in P$ containing $p'$. Is is easy to see that $\pi$ is an inclusion-preserving surjection. Function $g'$ is defined to be $g\pi$. The pair $\langle \pi, \mathrm{id} \rangle$ is clearly a morphism from $\mathcal{C}'$ to $\mathcal{C}$.

### 3.6 Integration of two classifications

In this section we work with regular classifications. (If the classifications are not regular, then we use the methods described above to regularize them. This process removes and merges only redundant elements, so nothing crucial is lost.) Suppose we are given two regular classifications, $\mathcal{C}_1 = \langle P_1, T_1, g_1 \rangle$ and $\mathcal{C}_2 = \langle P_2, T_2, g_2 \rangle$. From above, $\mathcal{C}_1 \cong \overline{\mathcal{C}}_1$ and $\mathcal{C}_2 \cong \overline{\mathcal{C}}_2$, and so we can work with the integration of the extensional forms. In fact, because of these isomorphisms, we will assume that $\mathcal{C}_1$ and $\mathcal{C}_2$ are already in extensional form, and so drop the overline symbol. The overall arrow diagram is below:

$$\begin{tikzcd}
T_1 \arrow[r, "\tau_1"] & T_1 \otimes T_2 & T_2 \arrow[l, "\tau_2"'] \\
P_1 \arrow[u, "g_1"] & P_1 \otimes P_2 \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[u, "g_1 \otimes g_2"'] & P_2 \arrow[u, "g_2"']
\end{tikzcd}$$

We begin by constructing the product partition $P_1 \otimes P_2$, defined in the usual way as:

$$P_1 \otimes P_2 = \{p_1 \cap p_2 | p_1 \in P_1, p_2 \in P_2, p_1 \cap p_2 \neq \emptyset\}$$

where $\pi_i$ maps $p \in P_1 \otimes P_2$ to the unique $p_i \in P_i$ containing $p$. The product partition $P_1 \otimes P_2$ is a refinement of both $P_1$ and $P_2$, and will be our integrated partition, upon which the fused classification is based. $P_1 \otimes P_2$ comes with two inclusion-preserving, surjective projection functions:

$$\pi_1 : P_1 \otimes P_2 \to P_1$$

$$\pi_2 : P_1 \otimes P_2 \to P_2$$

The classification function $g_1 \otimes g_2$ is defined by the rule,

$$g_1 \otimes g_2 : p_1 \cap p_2 \mapsto g_1 p_1 \cap g_2 p_2$$

The basis upon which $T_1 \otimes T_2$ is constructed is $\mathrm{Im}(g_1 \otimes g_2) \cup T_1 \cup T_2$. It may happen that this partial order is not a join semilattice, as two elements may have more than one minimal upper bound. The unique smallest lattice $L$ containing a partial order $P$ as a subset is a standard construction of lattice theory, called the Dedekind-MacNeille completion and written $L = \mathrm{DM}(P)$ (Grätzer, 1978). Therefore, the integrated taxonomy $T_1 \otimes T_2$ is constructed as:

$$T_1 \otimes T_2 = \mathrm{DM}(\mathrm{Im}(g_1 \otimes g_2) \cup T_1 \cup T_2)$$

The integrated classification is defined by

$$\mathcal{C}_1 \otimes \mathcal{C}_2 = \langle P_1 \otimes P_2, T_1 \otimes T_2, g_1 \otimes g_2 \rangle$$

The diagram makes clear that there are morphisms from $\mathcal{C}_1 \otimes \mathcal{C}_2$ to $\mathcal{C}_1$ and $\mathcal{C}_2$. We just need to observe that

$$g_1 \otimes g_2(p_1 \cap p_2) = g_1 p_1 \cap g_2 p_2 \subseteq g_1 p_1 = \tau_1 g_1 \pi_1(p_1 \cap p_2)$$

and

$$g_1 \otimes g_2(p_1 \cap p_2) = g_1 p_1 \cap g_2 p_2 \subseteq g_2 p_2 = \tau_1 g_2 \pi_2(p_1 \cap p_2)$$

### 3.7 Properties of the integration operation

This section considers some of the fundamental formal properties of the integration operation. As in the previous section, we assume that the classifications have already been pared and are therefore regular. We begin by constructing the following *unit* classification

$$\mathcal{U} = \langle P_{\mathcal{U}}, T_{\mathcal{U}}, g_{\mathcal{U}} \rangle$$

where $P_{\mathcal{U}}$ is the partition $\{X\}$, $T_{\mathcal{U}}$ is the join semilattice containing a single element, $\top$ say, and $g_{\mathcal{U}}$ maps $X$ to $\top$.

Suppose we are given two regular classifications, $\mathcal{C}_1 = \langle P_1, T_1, g_1 \rangle$ and $\mathcal{C}_2 = \langle P_2, T_2, g_2 \rangle$. Then, we have the following properties:

$$\mathcal{C}_1 \otimes \mathcal{C}_2 \cong \mathcal{C}_2 \otimes \mathcal{C}_1 \quad (1)$$

$$(\mathcal{C}_1 \otimes \mathcal{C}_2) \otimes \mathcal{C}_3 \cong \mathcal{C}_1 \otimes (\mathcal{C}_2 \otimes \mathcal{C}_3) \quad (2)$$

$$\mathcal{C}_1 \otimes \mathcal{U} \cong \mathcal{C}_1 \quad (3)$$

In other words, the collection of all classifications (strictly, all equivalence classes of classifications under classification isomorphism) has the algebraic structure of a monoid. Achieving these properties, which have a direct bearing on practical implementation of the construction, was a key goal during the development of the fusion construction. Together the commutative and associative properties (equations 1 and 2) ensure that for a set of classifications, the fusion construction will produce the same product irrespective of the order in which these classifications are integrated. Thus, the results of fusing two data classifications are suitable for inclusion in further fusion operations. The identity property (equation 3) ensures fusing a classification with the unit classification (the identity element) results in fusion product that is unchanged from the input classification.

### 3.8 Adding extra relations

The approach so far has been to allow the instance information of a classification to influence the classification structure. So, for example, if the total region classified as value $t$ is contained in the total region associated with class $t'$, then we infer that $t'$ subsumes $t$. We may additionally allow direct relationships between classifications to influence the structure of the integrated classification. This ability is important if we are integrating classifications in which we have not only instance information, but also direct knowledge about subsumption relations between elements of the two classification structures. The formal mechanism for constructing a new classification is that of a quotient structure.

Formally, let $T$ be a finite join-semilattice. We have noted above that a finite join-semilattice with $\perp$ is a lattice, so we can consider $T$ to be a lattice. Suppose, there is given an extra subsumption relationship $t \leq t'$, where $t, t' \in T$. We can rewrite the relationship as $t \wedge t' = t$ or $t \vee t' = t'$. Consider $t \wedge t' = t$. Construct a relation $\equiv'$ on $T$, where $x \equiv' y$ iff for some $z \in T$, $(x = z \wedge t$ and $y = z \wedge t \wedge t')$. Now, let $\equiv$ on $T$ be the reflexive, symmetric, transitive closure of $\equiv'$. It follows that $\equiv$ respects lattice meets, and so by basic lattice theory a quotient lattice $T/\equiv$ may be constructed, which is the lattice that takes account of our original subsumption relationship $t \leq t'$ in $T$.

Notice, that if we had begun with the equation $t \vee t' = t'$, we could have made a dual construction with a quotient lattice as result. However, this lattice may not be the same as the former quotient lattice. Indeed, we could use both the join and meet equations to get a third and possibly different result. The issue here is that we are revising our assumptions about subsumption relationships (given in $T$) based on new information. The way that this new information is incorporated will lead to differing results. Our choice of construction depends upon the context. We explore the implications of this result in the section 4.1.

## 4 Example application

To illustrate the fusion construction we provide a concrete example application, using two hypothetical land cover data sets. The aerial photograph in Figure 3 shows several different types of land cover in the region of Oldtown, Maine, USA.

We now suppose that two different individuals or organizations have produced digital land cover data for this region, using different ontologies. This situation is very common

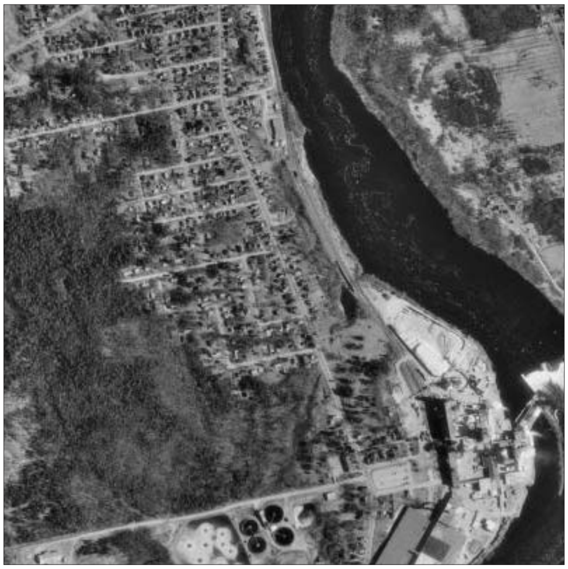


Figure 3: Aerial photograph of Oldtown, ME (Source: USGS aerial imagery)

for geographical information. Figures 4 and 5 show the taxonomies and the resulting land cover maps for these two data sets.

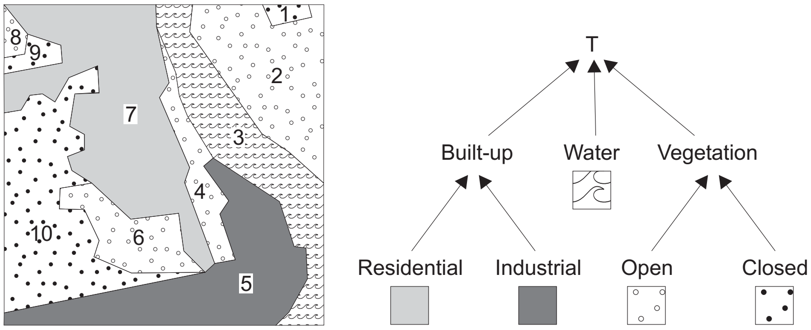


Figure 4: Land cover classification $\langle P_1, T_1, g_1 \rangle$ for Oldtown region

Referring back to the algebraic fusion construction in the previous section, the entire region is $X$; the land cover maps on the left-hand sides of Figures 4 and 5 are the partitions $P_1$ and $P_2$ of the region $X$; the taxonomies on the right hand sides of the same figures are the join semilattices $T_1$ and $T_2$. The relation between each taxonomy and the corresponding map is provided by the key (i.e. the shading patterns in both the map and the taxonomy) and is represented formally by $g_1$ and $g_2$. Note that only the leaf classes of a multi-level taxonomy are represented in the map (e.g. there are no regions labeled "Vegetation" in Figure 4). This is conventional for land cover data, but is not a requirement of the formal model. By labeling partition elements using the numbers in Figures 4 and 5 and labeling elements of each taxonomy using the first letter of the corresponding land cover class (so "Urban" becomes $U$), the two classifications can be expressed

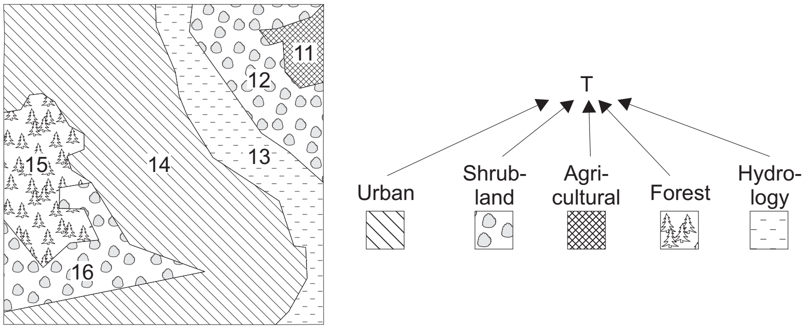


Figure 5: Land cover classification $\langle P_2, T_2, g_2 \rangle$ for Oldtown region

formally as:

$$
\begin{aligned}
&P_1 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}\\
&P_2 = \{11, 12, 13, 14, 15, 16\}\\
&T_1 = \{\top, B, C, I, O, R, V, W\}\\
&T_2 = \{\top, A, F, H, S, U\}\\
&g_1 : 1 \mapsto C, 2 \mapsto O, 3 \mapsto W, 4 \mapsto O, 5 \mapsto I, ..., 10 \mapsto C\\
&g_2 : 11 \mapsto A, 12 \mapsto S, 13 \mapsto H, 14 \mapsto U, 15 \mapsto F, 16 \mapsto S
\end{aligned}
$$

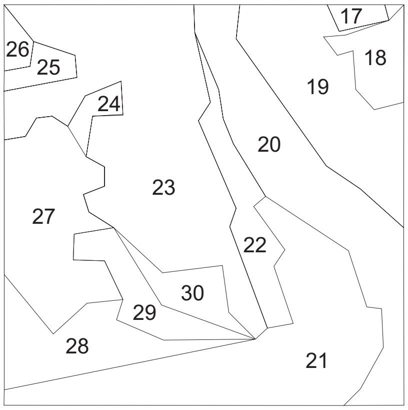


Figure 6: Product partition $P_1 \bigotimes P_2$

The product partition $P_1 \bigotimes P_2$ is shown in Figure 6. The two projection functions are given by:

$$
\begin{aligned}
&\pi_1 : 17 \mapsto 1, 18 \mapsto 2, 19 \mapsto 2, 20 \mapsto 3, 21 \mapsto 5, ..., 30 \mapsto 6\\
&\pi_2 : 17 \mapsto 12, 18 \mapsto 11, 19 \mapsto 12, 20 \mapsto 13, 21 \mapsto 14, ..., 30 \mapsto 14
\end{aligned}
$$

The next stage is to build the extension functions $e_1$ and $e_2$:

$$e_1 : \top \mapsto P_1, B \mapsto \{5,7\}, C \mapsto \{1,9,10\}, I \mapsto \{5\}, ..., W \mapsto \{3\}$$
$$e_2 : \top \mapsto P_2, A \mapsto \{11\}, F \mapsto \{15\}, H \mapsto \{13\}, ..., U \mapsto \{14\}$$

The extensional classifications $\overline{\mathcal{C}}_1 = \langle P_1, \overline{T_1}, \overline{g_1}\rangle$ and $\overline{\mathcal{C}}_2 = \langle P_2, \overline{T_2}, \overline{g_2}\rangle$ can now be defined, with $\overline{T_1} = e_1(T_1)$, $\overline{T_2} = e_2(T_2)$, $\overline{g_1} = e_1 g_1$, and $\overline{g_2} = e_2 g_2$. In this example, our input taxonomies $T_1$ and $T_2$ are already regular, so are isomorphic to $\overline{T_1}$ and $\overline{T_2}$. As a consequence, we again drop the overline notation in everything that follows.

The classification function $g_1 \otimes g_2$ is constructed as follows:

$$g_1 \otimes g_2 : 17 \mapsto \{1,9,10\} \cap \{12\}, 18 \mapsto \{2,4,6\} \cap \{11\}, ...$$

The integrated taxonomy is then constructed as the Dedekind-MacNeille completion of $\mathrm{Im}(g_1 \otimes g_2) \cup T_1 \cup T_2$, shown in Figure 7. For ease of reference, all of the taxonomies in Figure 7 has been relabeled with letters from the original classifications $T_1$ and $T_2$ replacing extensional labels. For example, from the definition above we know that $\{2,4,6\} \cap \{11\} = \{11\}$ is in $\mathrm{Im}(g_1 \otimes g_2)$. Consequently, in Figure 7 the extensional element $\{11\}$ has been relabeled as $A$ since $e_2(A) \mapsto \{11\}$.

Some new labels have been introduced to Figure 7 to label meets where no suitable label exists in $T_1$ or $T_2$ (e.g. $OU$ is the meet of $O$ and $U$). Figure 7 also shows all labels where classes are identified as equal (e.g. $W = H$). Note that a bottom element, $\perp$, and element $UV$ have been added to $\mathrm{Im}(g_1 \otimes g_2) \cup T_1 \cup T_2$ by the Dedekind-MacNeille completion to ensure $T_1 \otimes T_2$ is a lattice. Figure 8 illustrates the full fusion construction based on the commutative diagram at the beginning of 3.6, and annotated with example mappings for one element of the product partition (element 19).

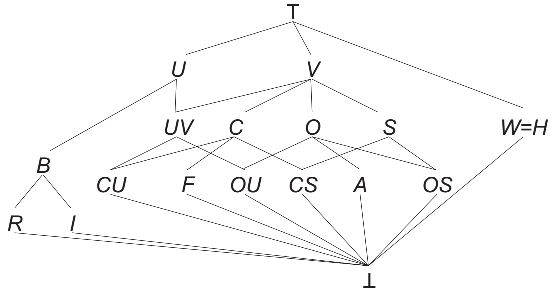


Figure 7: Integrated taxonomy lattice $T_1 \otimes T_2$

Finally, Figure 9 shows the fused data set in a conventional cartographic form.

## 4.1 Incorporating intensional knowledge

The fusion example discussed above is entirely instance-based and so can be completely automated (see section 6). As such it represents an important advance over conventional manual information fusion techniques. Based on extensional information about partonomic relationships between instances of land cover classes (e.g. “region 5 is part of region 14”), the fusion construction is able to infer taxonomic relationships between the

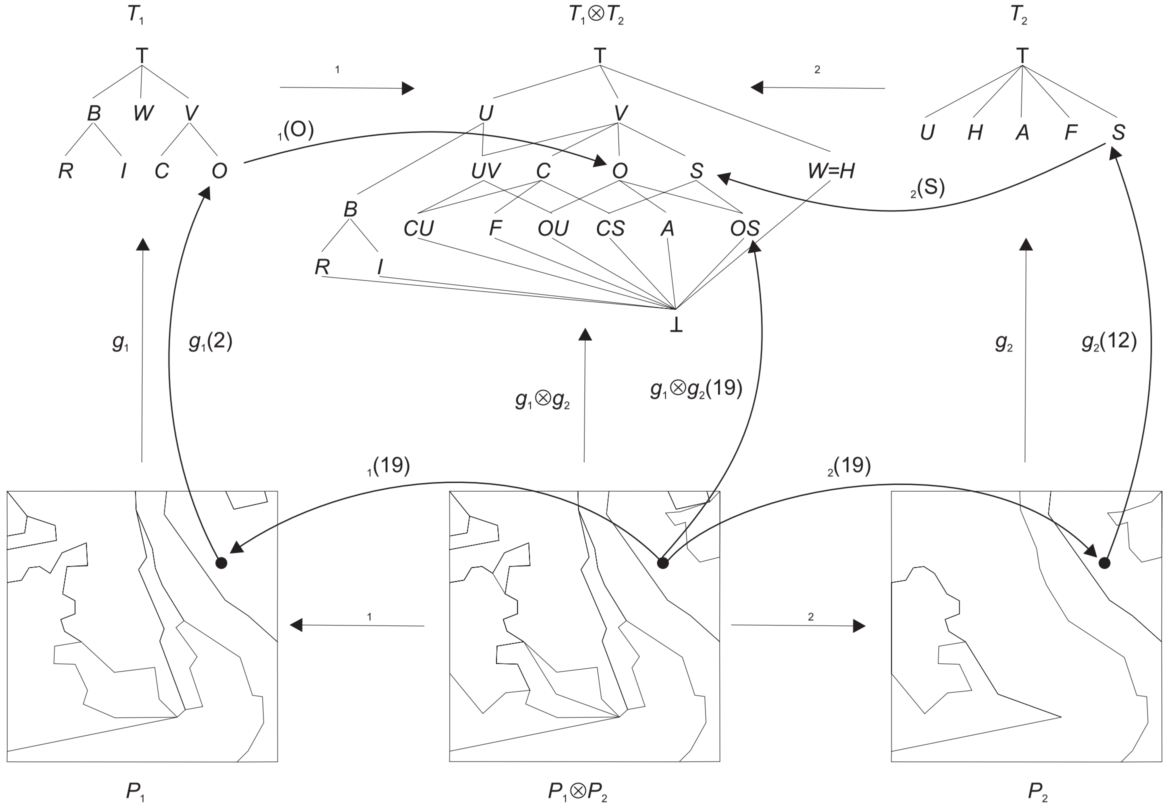


Figure 8: Full fusion construction, annotated with example mappings for product partition element 19

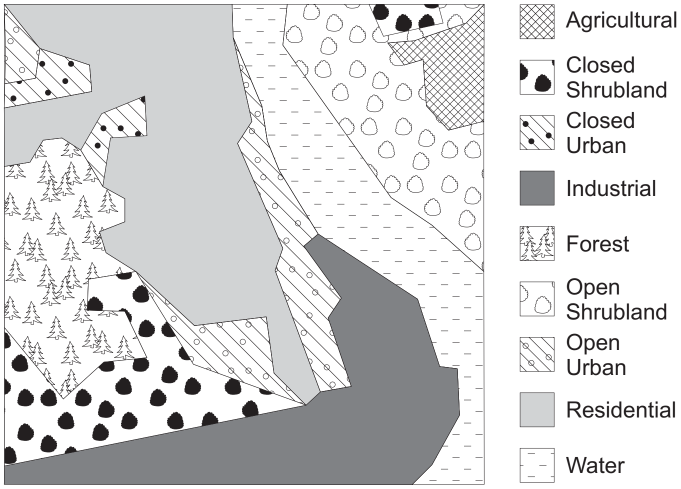


Figure 9: Fused land cover map

land cover classes themselves (e.g. "Industrial is subsumed by Urban"). There are many such taxonomic relationships shown in the derived taxonomy in Figure 7.

However, a central feature of the fusion construction is the ability optionally to include prior knowledge about the intensional relationships between taxonomies (see section 3.8). Returning to the land cover fusion example, assume we already possess intensional knowledge about the taxonomies $T_1$ and $T_2$ that the class $C$ ("Closed") is subsumed by the class $U$ ("Urban"). The quotient lattice induced by the equation $C \wedge U = C$ is shown in Figure 10. The dual construction of the quotient lattice induced by the equation $C \vee U = U$ is shown in Figure 11.

Some categories in Figure 10 have become identified with the bottom element $\perp$, specifically $F$ and $CS$. The quotient lattice in Figure 11 does not result in any such identifications. Instead, one category in Figure 11, $V$, has become identified with the top element $\top$. Which construction is most appropriate will depend on the balance of requirements for a particular application. The intuition behind the different structures is that the meet quotient lattice (Figure 10) produces results that are more *informative*, in the sense that categories can never become identified with more general categories. Conversely, the join quotient lattice (Figure 11) produces results that are more *conservative* in that categories can never become identified with more specific categories. Consequently, no category in the join quotient lattice can ever become associated with bottom, $\perp$.

## 5 Uncertainty

The fusion technique described above rests on the simplifying assumption that extensional information used to drive the inference process is perfect and certain. Imperfection is endemic in geographic information, as it is in many other types of information, and leads to uncertainty (Worboys and Duckham, 2004). Two important classes of imperfection in extensional information are *imprecision*, a lack of detail or specificity in information, and *inaccuracy*, a lack of correlation between information and the actual

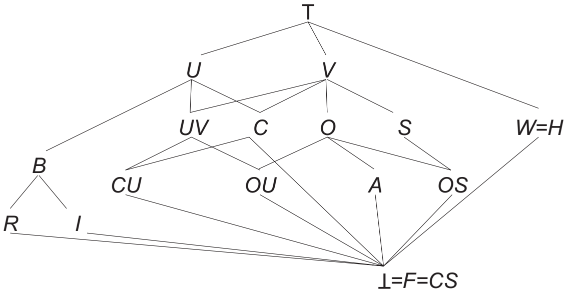


Figure 10: Revised taxonomy $T_1 \otimes T_2/\equiv$ incorporating prior knowledge $C \wedge U = C$

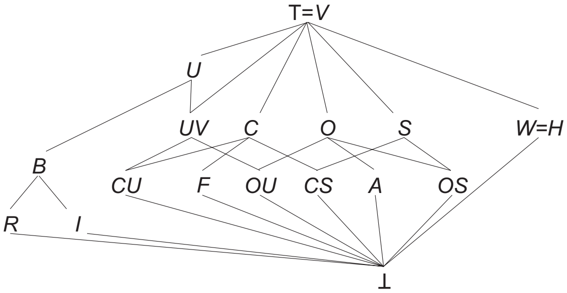


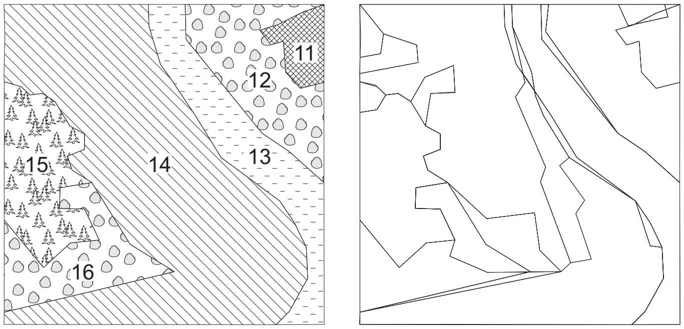
Figure 11: Revised taxonomy $T_1 \otimes T_2/\equiv$ incorporating prior knowledge $C \vee U = U$

state of affairs in the world (see Duckham et al., 2001).

For example, to simulate inaccuracy Figure 12a shows a slightly revised version of the partition in Figure 5, where some of the boundaries have been perturbed slightly. As a result of these perturbations, many additional polygons have been introduced into the product partition (Figure 12b), often referred to in the GIS literature as *sliver polygons* because of their long thin geometry.

The effect of the inaccuracy on the resulting fused taxonomy, as shown in Figure 13, is dramatic. This new taxonomy is much less useful that the original classification shown in Figure 7 as most regions are now simply classified by the meets of classes in the original classifications. In addition, the Dedekind-MacNeille completion introduces a further three new classes ($SV$, $UB$, and $UV$), required to ensure the result is a lattice. Comparing Figures 7 and 13, Figure 13 is clearly the more complex, and almost none of the subsumption relationships between classes shown in Figure 7 are to be found in Figure 13. Thus, the resulting construction is not really "fused" at all, since we have gained little new information about the intensional relationships between classes. Instead, the resulting construction is a degenerate fusion product closer to a conventional *overlay* of the two land cover data sets, a basic GIS operation for combining geospatial data. Figure 14 shows a "classification map" of the extension of resulting data set, where most of the extents have been classified using the meets of original classes (gray shading), and only one extent has been classified using a class in $T_1 \cup T_2$ (white shading).

Addressing the problem of degenerate fusion products is important to the success of automated information fusion. The problem will be a key area of future research, but in



a. Inaccuracy simulation b. Revised product partition

Figure 12: Inaccuracy in extensional information

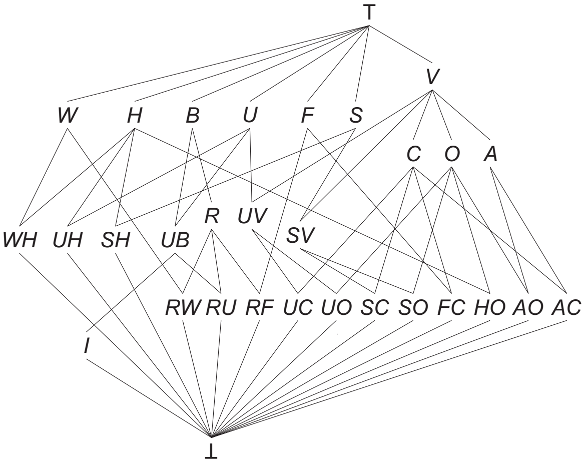


Figure 13: Degenerate taxonomy resulting from disregarding inaccuracy in extensional information

this paper we present one solution that has the advantages of being both simple and highly effective. The sliver polygons that result from inaccuracy are often easy to identify, either by their relatively small area or elongated geometry. Once identified, sliver polygons can then be omitted from the product partition. For example, the degree to which $p_2 \in P_2$ overlaps $p_1 \in P_1$ can be calculated as $o(p_2, p_1) = \text{area}(p_1 \cap p_2)/\text{area}(p_1)$. Similarly, the degree to which $p_1$ overlaps $p_2$ can be calculated as $o(p_1, p_2) = \text{area}(p_1 \cap p_2)/\text{area}(p_2)$. Finally, for some threshold value $k_o \in [0, 1]$, we can redefine the product partition, $P_1 \bigotimes P_2$ in equation 4, as:

$$\{p_1 \cap p_2 | p_1 \in P_1, p_2 \in P_2, p_1 \cap p_2 \neq \emptyset, o(p_1, p_2) \leq k_o \text{ and } o(p_2, p_1) \leq k_o\} \quad (4)$$

In the case of our land cover example, setting the threshold $k_o$ to about 0.1 (i.e. ignoring any polygon intersections with an area of less than 10% of both the two overlaping polygons) allows us to regain the original fused taxonomy in Figure 7. The cost of ignoring these small overlaps is that some of the sliver polygons will be unclassifiable in

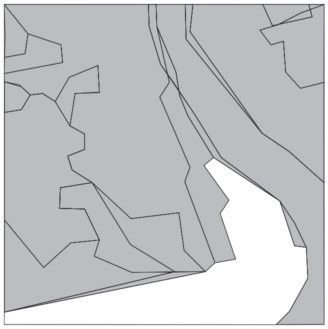


Figure 14: Degenerate classification map, showing fused extents where $t \in T_1 \cup T_2$ (white shading) or $t \in \mathrm{Im}(g_1 \otimes g_2)\backslash T_1 \cup T_2$ (gray shading)

the fused taxonomy. The total area of such unclassifiable extents can only ever be as great as the threshold $k_o$ (i.e. 10% in our example), and in practice it will be much lower (approximately 3% in our example).

There is, therefore, a balance to be struck between the quality of extensional and intensional information in the fusion construction under inaccuracy. Tolerating higher levels of inaccuracy leads to higher quality, more useful intensional information, but lower quality extensional information with more unclassifiable regions. Conversely, tolerating lower levels of inaccuracy leads to lower quality, less useful intensional information, but higher quality extensional information with fewer unclassifiable regions. The revised classification map resulting from a $k_o$-threshold of 0.1 is shown in Figure 15. When compared with 14 it is clear that many fewer locations have been classified using the meets of classes in the original classifications (gray shading), although some locations are now unclassifiable (hatched shading).

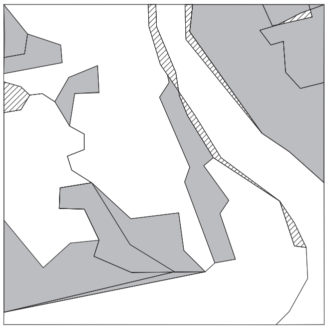


Figure 15: Classification map using $k_o$-threshold of 0.1, showing fused extents where $t \in T_1 \cup T_2$ (white shading), $t \in \mathrm{Im}(g_1 \otimes g_2)\backslash T_1 \cup T_2$ (gray shading), or where extent is unclassifiable (hatched shading)

Imprecision can be addressed in a similar fashion. Another threshold $k_g \in [0, 1]$ can be used to exclude those extents where $o(p_1, p_2) \leq k_g$ and $o(p_2, p_1) \geq 1 - k_g$ from the product partition. The intuition behind this formula is to exclude fine grained detail where a very large proportion of one extent overlaps a very small portion of another extent. In the following section we explore one technique for finding the most desirable threshold values.

## 6 Implementation and testing

The examples described above have all been tested within a prototype implementation. In this section we briefly describe this implementation.

Rather than translate the algebraic construction into a procedural language, like Java, it was much more efficient from a conceptual perspective to implement this construction directly within a pure functional programming language. Thus, the core code of the fusion construction was written in the functional programming language Scheme (Abelson and Sussman, 1996). Using Scheme allows most of the fusion construction to be programmed directly as explained here, with only minor syntactic changes. For example, the definition of the function $\overline{g_1}$, discussed in section 3.3 ($\overline{g_1} : p \mapsto e_1 g_1 p$) can be written directly within Scheme as:

```
(define g1bar (lambda (p) (e1 (g1 p))))
```

where the Scheme functions `g1` ($g_1$) and `e1` ($e_1$) are defined elsewhere in a similar fashion. There exists a direct correspondence between the Scheme program fragment and the definition of the function $\overline{g_1}$. The `lambda` construct in functional programming is derived from lambda calculus (see Barendregt, 1984) and binds occurrences of `p` in the head of the function definition to occurrences in the body of function.

Scheme code was used to implement all of the fusion construction up to the computation of the Dedekind-MacNeille completion, which was implemented in Java using an algorithm based on Bertet et al. (1997). Finally, the user interface and all remaining code were implemented within Java. An open-source Java API for Scheme (Kawa) enabled the entire system to be executed as a single Java application. The overall architecture is summarized diagrammatically in Figure 16.

The data import subsystem allows geospatial data to be imported in the common .shp Shapefile format. In addition to the extensions of the geospatial data within the Shapefile, a description of the intensional subsumption relationships within the taxonomy is also needed. In our prototype example, the taxonomy was expressed directly within Scheme. In a practical system, the data and taxonomy would be available to the fusion system via standard queries to the geospatial database system. The Java application middleware then communicates this information to the Kawa Scheme fusion engine, which returns the fused taxonomy. After computing the Dedekind-MacNeille completion of the taxonomy, this taxonomy is then displayed to the user, along with a classification map of the integrated data.

The discussion in the previous sections highlighted the importance of $k$-thresholds in effective fusion under uncertainty. The $k$-threshold levels are controlled by the user via two sliders, shown in Figure 16, one for the overlap tolerance ($k_o$), the other for the granularity tolerance ($k_g$). Setting the sliders to appropriate levels is the only user interaction needed by the fusion process, other than to the import relevant data sets. Increasing the $k$-threshold tolerances increases both the level of integration in the fused taxonomy and the

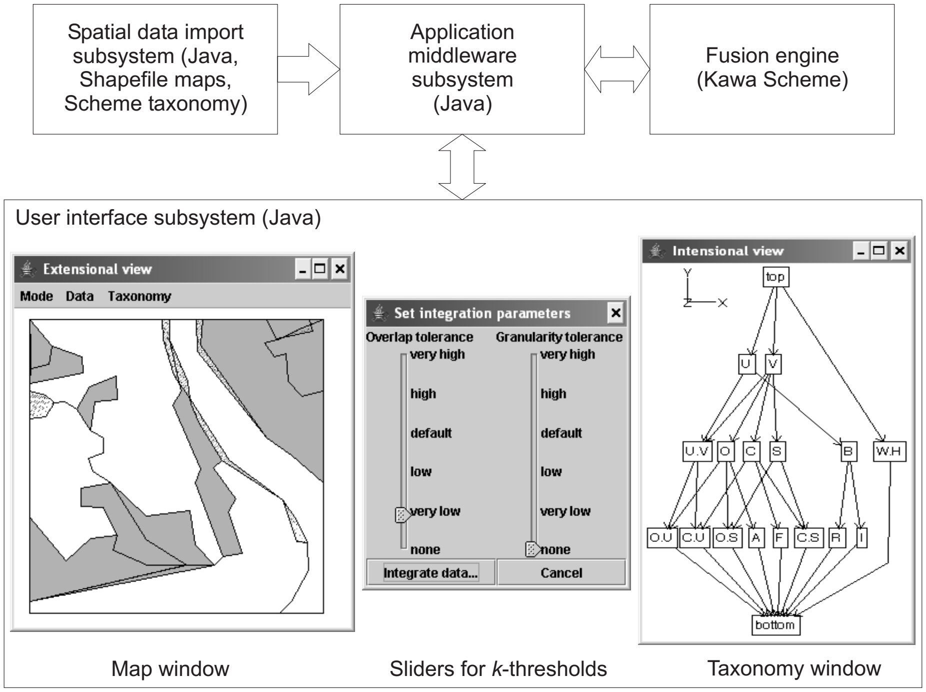


Figure 16: System architecture overview

level of unclassifiable regions in the fused extents. The interface provides rapid real-time feedback on the fusion, so users are able to experiment with different $k$-thresholds and interactively view the results both in terms of the fused taxonomy and the classification map.

In addition to fabricated data sets, like those shown in this example, the system has also been tested with several real land-cover data sets, such as USGS land use/land cover data and local government land cover data in the US and Ordnance Survey MasterMap data and CORINE land cover data for regions in the UK. In the following section we discuss the future research that needs to be conducted before the approach can be considered for practical applications, although initial results indicate that the prototype implementation is able to successfully fuse such data.

## 7 Discussion and conclusions

This paper has set out a new technique for automating the process of information fusion. This technique is founded on a formal algebraic model of information fusion that models both extensional and intensional aspects of information. Extensional relationships between input data sets are used to automatically infer intensional relationships in the fused data set, an inference process that would usually require human domain expertise. This automatic inference is the key feature that distinguishes our information fusion construction from a simple overlay procedure. Crucially for a fusion process, the algebraic fusion construction has the properties of a monoid (section 3.7). Thus, for a set of classifications, associativity and commutativity ensure that the results of the fusion process will be unaffected by the order in which classifications are fused.

Where prior knowledge of intensional relationships already exists, this information can also be included in the fusion process. Uncertainty in the input data sets, such as extensional inaccuracy or imprecision, can have a significant effect upon the inference process. Under uncertainty, a balance exists between the quality of extensional and intensional information in the fused data set. Higher quality extensional information with low proportions of unclassifiable extents can only be achieved at the cost of lower quality intensional information with poor integration between the input schema. Conversely, tolerating lower quality extensional information with some unclassifiable extents leads to much higher quality integration of input schema. The formal model has been successfully tested and implemented, using a combination of procedural programming and functional programming.

## 7.1 Further work

Two important areas of further work have already been suggested in the paper. First, the use of $k$-thresholds to set tolerance levels for imprecision and inaccuracy is both simple and effective. However, more research is needed into other approaches to uncertainty to determine whether more effective techniques might also be developed. Second, so far only subjective assessments of the fusion products have been attempted. Specifically, future work will need to:

- Assess objectively the fidelity of fusion results, and verify that fusion products are indeed suitable for use within different application areas.
- Compare the prototype automated fusion system with conventional manual fusion processes, and verify from an HCI perspective that the automated fusion system is indeed easier to use.

In addition to these topics, several other areas yet to be explored are suggested by this research. First, the technique has been described here as a system for fusing semantically related information sources, such as land-cover data sets. However, the same technique might equally be used as a data mining system for discovering relationships between semantically unrelated information sources. For example, if instead of two land cover data sets the algebraic construction is applied to one land cover data set and one socioeconomic data set. If the resulting taxonomy indicates that, say, the “Agricultural” land cover class is subsumed by the “Low per-capita income” socioeconomic class, this might suggest some form of causal relationship between the two classes (even though it would not be true to say that the concept “Agricultural” is a sub-concept of “Low per-capita income”).

Second, the prototype implementation used user-defined $k$-thresholds to set the correct balance between intensional and extensional information fusion. In fact, there are a variety of possible ways in which the selection of $k$-thresholds could be automated. For example, by iterating the fusion process using different $k$-thresholds it should be possible to find those thresholds that optimize the fusion for classifiability (i.e. the lowest thresholds that result in no unclassifiable regions). Conversely, by developing a formal notion of how informative an integrated taxonomy is (for example, using measures of semantic similarity as in Rodríguez and Egenhofer, 2003), it might also be possible to optimize the fusion for information content (i.e. the highest thresholds that result in maximal information content in the integrated taxonomy).

Finally, the fusion construction has been explored within the domain of geospatial information, because a common geographic coordinate system provides an ideal structure for driving the inference process. However, as mentioned previously the extensional structure does not necessarily need to be spatial. Consequently, future research will also focus on the applicability of the information fusion technique to a variety of non-spatial application domains. For example, the set of results returned be a WWW search engine in response to a particular query term can be thought of as a "region" of the WWW and as the extension of that query term. Different "regions" of the WWW may "overlap," in the sense that they share related URLs or refer to pages with similar text. This non-spatial structure between extensional "regions" of WWW query terms can be used to drive the same information fusion process as described in this paper. This approach is currently being explored as the basis for further related research into information fusion within the semantic web.

## Acknowledgments

The authors are grateful to John Stell and Nic Wilson for several helpful discussions that contributed to the development of this work. The thorough and thoughtful comments from the three anonymous reviewers were also helpful.

## References

Abelson, H. and Sussman, G. J. (1996). *The structure and interpretation of computer programs*. MIT Press, Cambridge, MA, 2nd edition.

Barendregt, H. P. (1984). *The Lambda Calculus*. North-Holland.

Berlin, J. and Motro, A. (2001). Autoplex: Automated discovery of contents for virtual databases. In *Proceedings of COOPIS 2001, Sixth IFCIS International Conference on Cooperative Information Systems*, volume 2172 of *Lecture Notes in Computer Science*, pages 108–122, Berlin. Springer.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 279(5):34–43.

Bertet, K., Morvan, M., and Nourine, L. (1997). Lazy completion of a partial order to the smallest lattice. In *International KRUSE Symposium: Knowledge Retrieval, Use and Storage for Efficiency*, pages 72–81, University of Vancouver.

Brodaric, B. and Gahegan, M. (2002). Distinguishing instances and evidence of geographical concepts for geospatial database design. In Egenhofer, M. J. and Mark, D. M., editors, *Geographical Information Science*, volume 2478 of *Lecture Notes in Computer Science*. Springer.

Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., and Rosati, R. (1998). Description logic framework for information integration. In *Proceedings 6th International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 2–13.

Doan, A., Domingos, P., and Levy, A. Y. (2000). Learning source description for data integration. In *WebDB (Informal Proceedings)*, pages 81–86.

Doan, A., Madhavan, J., Domingos, P., and Halevy, A. (2002). Learning to map between ontologies on the semantic web. In *WWW 2002, Proceedings 11th International World Wide Web Conference*, pages 662–673. ACM.

Duckham, M., Mason, K., Stell, J. G., and Worboys, M. F. (2001). A formal approach to imperfection in geographic information. *Computer, Environment, and Urban Systems*, 25:89–103.

Fonseca, F., Davis, C., and Camara, G. (2003). Bridging ontologies and conceptual schemas in geographic information integration. *Geoinformatica*, 7(4):355–378.

Fonseca, F., Egenhofer, M., Agouris, P., and Câmara, G. (2002). Using ontologies for integrated geographic information systems. *Transactions in GIS*, 6(3):231–257.

Fonseca, F. T. and Egenhofer, M. J. (1999). Ontology-driven geographic information systems. In Medeiros, C. B., editor, *Proceedings Seventh Symposium on Advances in Geographic Information Systems*, pages 14–19.

Ganter, B. and Wille, R. (1999). *Formal Concept Analysis*. Spinger, Berlin.

Grätzer, G. (1978). *General Lattice Theory*. Academic Press, New York.

Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220.

Guarino, N. (1998). Formal ontology and information systems. In *Formal Ontology and Information Systems, Proceedings FOIS'98*, pages 3–15. IOS Press.

Guarino, N., Masolo, C., and Vetere, G. (1999). Ontoseek: Content-based access to the Web. *IEEE Intelligent Systems*, 14(3):70–80.

Kavouras, M. and Kokla, M. (2002). A method for the formaizlation and integration of geographical categorizations. *International Journal of Geographical Information Science*, 16(5):439–453.

Kim, W. and Sea, J. (1992). Classifying schematic and data heterogeneity in multidatabase systems. *IEEE Computer*, 24(12):12–18.

Kokla, M. and Kavouras, M. (2001). Fusion of top-level and geographic domain ontologies based on context formation and complementarity. *International Journal of Geographical Information Science*, 15(7):679–687.

Lakshmanan, L., Sadri, F., and Subramanian, I. (1993). On the logical foundations of schema integration and evolution in heterogeneous database systems. In Ceri, S., Tanaka, K., and Tsur, S., editors, *DOOD '93, Proceedings Third International Conference on Deductive and Object-Oriented Databases*, volume 760 of *Lecture Notes in Computer Science*, pages 81–100. Springer, Berlin.

Li, W.-S. and Clifton, C. (1994). Semantic integration in heterogeneous databases using neural networks. In Bocca, J. B., Jarke, M., and Zaniolo, C., editors, *VLDB'94,Proceedings 20th International Conference on Very Large Data Bases*, pages 1–12. Morgan Kaufmann.

Li, W.-S. and Clifton, C. (2000). SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data and Knowledge Engineering*, 33:49–84.

Rahm, E. and Bernstein, P. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350.

Rodríguez, A. and Egenhofer, M. (2003). Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):442–456.

Sheth, A. (1999). Interoperability and spatial information theory. In Goodchild, M., Egenhofer, M., Fegeas, R., and Kottman, C., editors, *Interoperating Geographic Information Systems*, chapter 2, pages 5–29. Kluwer, Dordrecht, Netherlands.

Sheth, A. and Kashyap, V. (1993). So far (schematically) yet so near (semantically). In Hsiao, D., Neuhold, E., and Sacks-Davis, R., editors, *DS-5, Proceedings IFIP Database Semantics Conference on Interoperable Database Systems*, volume 25 of *IFIP Transactions*, pages 283–312. North-Holland.

Tzitzikas, Y., Spyratos, N., and Constantopoulos, P. (2001). Mediators over ontology-based infomation sources. In *Proceedings WISE 1*, pages 31–40.

Ullman, J. (2000). Information integration using logical views. *Theoretical Computer Science*, 239(2):l89–210.

Vckovski, A. (1998). *Interoperable and Distributed Processing in GIS*. Taylor & Francis, London.

Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., and Hübner, S. (2001). Ontology-based integration of information—a survey of existing approaches. In Stuckenschmidt, H., editor, *IJCAI-01 Workshop: Ontologies and information sharing*, pages 108–117.

Wald, L. (1999). Definitions and terms of reference in data fusion. In Baltsavias, E., Csatho, B., Hahn, M., Koch, B., Sieber, A., Wald, L., and Wang, D., editors, *International Archives of Photogrammetry and Remote Sensing*, volume 32, pages 2–6.

Widom, J. (1995). Research problems in data warehousing. In *Proceedings 4th International Conference on Information and Knowledge Management (CIKM)*.

Wiederhold, G. (1992). Mediators in the architecture of future information systems. *IEEE Computer*, 25(3):38–49.

Worboys, M. F. and Duckham, M. (2002). Integrating spatio-thematic information. In Egenhofer, M. J. and Mark, D. M., editors, *Geographic Information Science*, volume 2478 of *Lecture Notes in Computer Science*, pages 346–361. Springer, Belin.

Worboys, M. F. and Duckham, M. (2004). *GIS: A Computing Perspective*. CRC Press, Boca Raton, FL, 2nd edition.

Zhou, G., Hull, R., King, R., and Franchitti, J.-C. (1995). Data integration and warehousing using h2o. *IEEE Data Engineering Bulletin*, 18(2):29–40.