

Semi-Automatic Ontology Alignment for Geospatial Data Integration*

Isabel F. Cruz, William Sunna, and Anjali Chaudhry

Department of Computer Science
University of Illinois at Chicago
851 S. Morgan St. (M/C 152), Chicago, IL 60607, USA
[ifc,wsunna,achaudhr]@cs.uic.edu

Abstract. In geospatial applications with heterogeneous databases, an ontology-driven approach to data integration relies on the alignment of the concepts of a central ontology that describe the domain, with the concepts of the ontologies that describe the data in the local databases. Once the alignment between the global ontology and each local ontology is established, users can potentially query hundreds of databases using a single query that hides the underlying heterogeneities. Using our approach, querying can be easily extended to a new data source by aligning a local ontology with the global one. For this purpose, we have designed and implemented a tool to align ontologies. The output of this tool is a set of mappings between concepts, which will be used to produce the queries to the local databases once a query is formulated on the global ontology. To facilitate the user's task, we propose semi-automatic methods for propagating such mappings along the ontologies. In this paper, we present the principles behind our propagation method, the implementation of the tool, and we conclude with a discussion of interesting cases and proposed solutions.

1 Introduction

In an ontology-driven approach, an application that needs to use the integrated data from a domain expresses its information requests in terms of the concepts in the global ontology, thus giving users the appearance of an homogeneous view over heterogeneous data sources.

An ontology ranges from a simple taxonomy to an axiomatized set of concepts and relationship types [9]. In our approach, we focus on taxonomies for land use coding in the state of Wisconsin. Our work is in collaboration with the Wisconsin Land Information System (WLIS), which is a distributed web-based system with heterogeneous data residing on local and state servers.

In order to develop WLIS, it is necessary to overcome data heterogeneity that originates from having different state and federal agencies involved in acquiring and storing geospatial data. On one hand, proposing a standard ontology to be adopted by all agencies would lead to loss in the resolution of the collected data. A more feasible approach is to allow for the different agencies to maintain their own ontologies locally while specifying how concepts in their own ontology correspond to concepts in a global ontology. On the other hand, we note that even when agencies use a particular standard, their exact implementation of that standard will differ. In the presence of such heterogeneities, however small they may be, queries to one ontology will likely not work on another ontology.

We focus primarily on *ontology alignment*, that is, on establishing mappings between related concepts in the global ontology and a local ontology without combining the ontologies. We also discuss *ontology merging* when the two ontologies are not only aligned but also included in a single, coherent ontology.

In order to integrate the constituent data models, the mappings between concepts in the global ontology and those in the local data sources, e.g., as described by a local ontology, have to be determined. In our approach, the mappings are determined semi-automatically, that is, partly established manually by the user and partly deduced using an automatic process.

The paper is organized as follows. In Section 2, we give an overview of related work in the area. We present the land use codes and how we encode them using an ontology in Section 3. In Section 4 we present the mappings we consider in performing the alignment between two ontologies, we describe our approach to

* This research was supported in part by the National Science Foundation under Awards EIA-0091489 and ITR IIS-0326284.

automatically propagate mappings along the two ontologies involved in the mapping, state the assumptions we make that are necessary for the automatic propagation to work, and give an example on how alignments can be used for merging ontologies. In Section 5 we describe the implementation of the mapping tool that supports the semi-automatic deduction process. With the ultimate goal of extending our semi-automatic alignment method to any ontologies, in Section 6 we give examples of ontologies that do not satisfy our previously made assumptions and propose ways of extending our semi-automatic propagation to handle those new cases. Finally, in Section 7, we draw conclusions and outline future work.

2 Related Work

Hovy [4] proposes a semi-automatic ontology alignment approach for combining and standardizing large ontologies based on their lexical similarity. Multiple languages are also supported by consulting dictionaries. Different kinds of matches, including text and hierarchy matches are used in the heuristics for the alignment process. Initially the unaligned ontologies are loaded and brought into a partially aligned state. For all unaligned concepts, the heuristics mentioned earlier are used to create a set of cross-ontology match scores. Then, a new set of alignment suggestions are created by a function that combines the match scores. The user can then check the suggestions and retain the best matches.

Chimaera [5] is a software tool developed by the KSL group at Stanford, also used for merging and testing large ontologies. It provides tools for merging different ontologies created by different authors and for diagnosing individual or multiple ontologies. Similarly to Hovy’s approach, the tool allows users to check the automated merging procedure. Classes that need the user’s attention are highlighted. PROMPT [7], which also uses a frame-based model for the ontology, supports merges at the slot level, in addition to merges at the class (frame) level.

In the ontologies that we have considered so far for land use management, an ontology is a hierarchy where concepts refer to the codes (the vertices in the hierarchy) and relationships are established between a parent code and a child (the edges between the corresponding vertices). Such relationships represent generalization/specialization between the codes. In our case, there are no explicitly represented properties or attributes associated with the codes. We have a simpler structure than that found in other systems [5, 7], and therefore the decision of whether two concepts match has to be solely based on the codes.

Another system for the alignment of ontologies proposes a comprehensive approach encompassing a rich set of mappings types [8]. The authors stress the importance of declaratively specified mappings (our mappings are declarative), the necessity for automation, and of addressing the problem of defining a measure for the expressiveness of the mappings supported. However, several of these issues are left as future work.

The MOMIS approach [1, 2] has several similarities with our approach. It include the semi-automatic processing of ontology alignment, but it has several different characteristics. It uses subsumption and lexical comparison, but in our case we have strings not frames therefore subsumption is not a possibility because we have only strings to consider instead of frames, and lexical comparison, as that provided by WordNet [6] would only play a limited role in the types of ontologies we currently consider.

3 Land Use Data

Our examples focus on the Wisconsin Land Information System (WLIS) and on Land Use Data. A land use database system stores information about land parcels in XML format. Our land parcel data contains an identification number for the parcel (represented by the tag *lid*), the category of land usage under which it is classified (*lucode*), the file containing the pertinent shape information (*shape.file*), and information about the owner of the parcel (*owner.id*).

Land use categories include *agriculture*, *commerce*, *industry*, *institutions*, and *residences*. Storing the land use codes of land parcels helps in better planning for township development, transportation, taxation, and so on. A typical query such as “Where are all the crop and pasture lands in Dane County?” would be relatively straightforward when using one data set but more difficult when posed over a larger geographic area. Table 1 illustrates the heterogeneity of attribute names and values that would satisfy the criteria of the query over selected multiple data sets.

Table 1. Heterogeneity of attribute names and values

Planning Authority	Attribute	Code	Description
Dane County RPC	Lucode	91	Cropland Pasture
Racine County (SEWRPC)	Tag	811	Cropland
		815	Pasture and Other Agriculture
Eau Claire County	Lu1	AA	General Agriculture
City of Madison	Lu_4_4	8110	Farms

1. *Lucode*, *Tag*, *Lu1* and *Lu_4_4* must be resolved as synonyms for the attribute that represents the land use code in the ontology.
2. The descriptions are not exact matches. For example where one code is used for the remaining classifications, Racine County uses two codes.

To represent the ontologies we use our own XML DTD [3].

4 Ontology Alignment

An important step in the data integration process is ontology alignment—the identification of semantically related entities in different ontologies. While establishing a semantic relationship between concepts in the global ontology and concepts in the local ontologies can be challenging, the thorough identification of such relationships is essential for the development of accurate machine-based techniques to handle them. Aligning very heterogenous ontologies can be a difficult process. In our approach we expect ontologies to be close to each other in a given domain.

In the examples, we represent the ontologies as trees. The vertices of the trees correspond either to existing entities in the ontology (*real vertices*) or to entities created with the end of logically grouping entities (*virtual vertices*). In the figures, the left tree represents the global ontology and the right tree represents the local ontology.

For example, in Figure 1, the codes *Agriculture - Woodlands - Forests* and *Agriculture - Woodlands - Non-forests* in the ontology are mapped to the land use codes *Forestry (91)* and *Non-forest woodlands (92)* in the local ontology (used by Dane County in Wisconsin). There is no local land use code corresponding to *Agriculture - Woodlands*. The land use codes of the land parcels in the database are stored only as *91* or *92*, corresponding to *Forestry* and *Non-forest woodlands*. To better align the local ontology with the ontology, a virtual vertex was introduced corresponding to *Agriculture - Woodlands*.

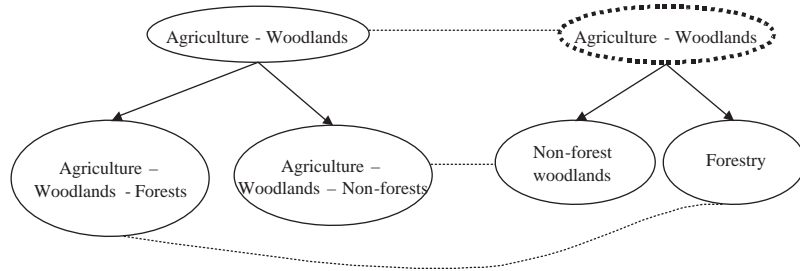


Fig. 1. Real and virtual vertices

4.1 Mapping Types

In Figure 1, *Agriculture Woodlands - Forests*, *Agriculture Woodlands - Non-forests*, *Forestry* and *Non-forest woodlands* are semantically at the same level of detail in the two ontologies. Similarly, the two vertices

corresponding to *Agriculture - Woodlands* also at the same level. We say that such concepts are *aligned*. Initially, the information as to which entities in the different ontologies are aligned must be provided by the user, who is the geospatial expert for the local database. Once two entities are known to be aligned, the nature of the relation between them can be characterized using the following mapping types: *Exact*, the connected vertices are semantically equivalent, *Approximate*, the connected vertices are semantically approximate, *Null*, the vertex in the ontology does not have a semantically related vertex in the local ontology, *Superset*, the vertex in the ontology is semantically a superset of the vertex in the local ontology, and *Subset*, the vertex in the ontology is semantically a subset of the vertex in the local ontology.

A mapping can establish the connection between vertices in their entirety or only to parts of a vertex, based on the semantics. Even though the global ontology is usually developed by a team of domain experts, who take utmost care in making sure that every semantically unique entity is represented by exactly one vertex, the local ontology might consist of entities organized or grouped using different criteria. As a result, the semantic equivalent of an entity in the global ontology could be distributed over several vertices or parts of a vertex in the local ontology and vice versa. A county in which agriculture is the main occupation will have more categories of agricultural land usage than the ontology drawn up for the state. Such differences in the resolution of the data can also lead to complications in ontology alignment.

Figure 2 illustrates several mappings between vertices in two ontologies for land use patterns in a centralized integrated system. We show the global ontology subtree on the left side of figure and the local ontology subtree on the right. The vertices corresponding to *Industry*, *Mining* and *Manufacturing* in the global ontology can be mapped to those corresponding to *Industrial Sector*, *Mining* and *Production* in the local ontology. In the global ontology, the vertex *Plastic wares* denotes entities that are made of plastic or glass. However, in the local ontology, there is a vertex *Plastics* and another vertex *Rubber and Glass*, which denotes manufactured objects made of rubber or glass.

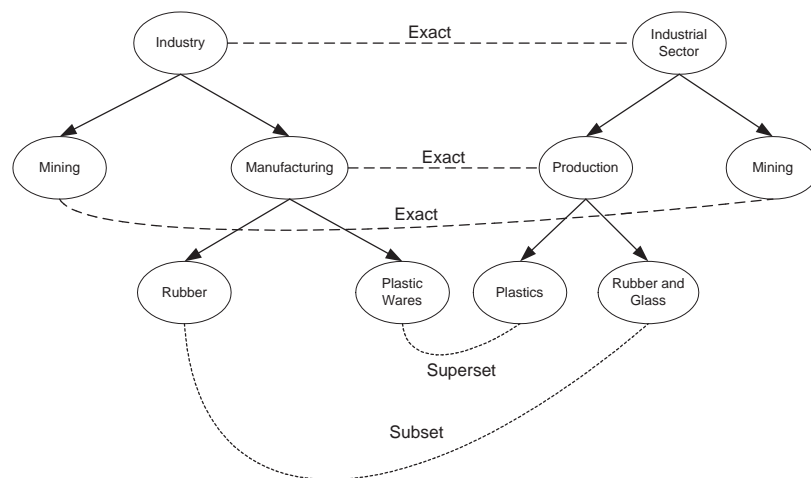


Fig. 2. Mapping types.

The *Manufacturing* and the *Production* vertices are aligned. Similarly, the two *Mining* vertices are also aligned. *Manufacturing* is semantically equivalent to *Production*, as both denote a collection of industries producing plastics, glass, and rubber products. Hence, this mapping is of type *Exact* as denoted in the mapping from the *Manufacturing* vertex to the *Production* vertex. *Plastic wares* is semantically a *superset* of the *Plastics* vertex and *Rubber* is semantically a *subset* of the *Rubber and Glass* vertex.

4.2 Semi-automatic Alignment

To allow for the semi-automatic processing of the ontology alignment, we propose a framework that defines the values associated with the vertices of the ontology in two possible ways: as functions of the values of the children vertices or of the user input.

We need to establish two assumptions to guarantee the correctness of the deduction process. The first one is that the specialization of a vertex in the ontology must be total, that is, each lower-level concept must belong to a higher-level concept in the hierarchy. The second one is that “bowties” [4], which are inversions in the order of the two ontologies that are being aligned, do not occur.

The mapping techniques described in Section 4.1 can be integrated in a semi-automatic alignment methodology to simplify the task of aligning large ontologies [4]. The user initially identifies the hierarchy levels in the two ontologies that are aligned. Then the alignment component propagates. When ambiguities or inconsistencies are encountered, or the the algorithm can not propagate values any further, those vertices are singled out. The user can then manually assist the algorithm by mapping concepts by hand.

For example, in Figure 3, vertex b in the global ontology is mapped using mapping type *Superset* to vertex e in the local ontology, and vertex c in the global ontology is mapped using mapping type *Exact* to vertex f in the local ontology. The mapping type between their parents a and d can be deduced to be *Superset* based on the mapping between the children, because we consider that the semantic content of the parent is the aggregation of the semantic contents of its children.

All the children of d are mapped to children of a . This is the *Fully Mapped (FM)* case. The *Partially*

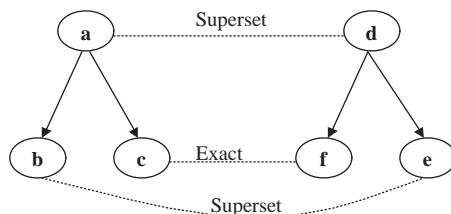


Fig. 3. Fully mapped deduction operation

Mapped (PM) case occurs if there are some children in the local ontology that cannot be mapped to any of the children in the global ontology. For example, in Figure 4 vertices b and c in the global ontology are mapped to vertices e and f using mapping type *Exact*. But vertex g cannot be mapped to any of the vertices in the global ontology. As a result, vertex a is mapped using mapping type *Subset*.

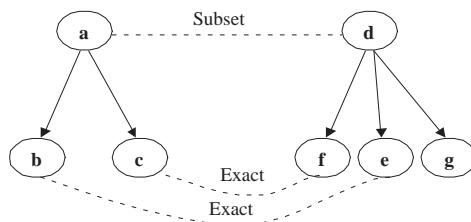


Fig. 4. Partially mapped deduction operation

Table 2 lists the different possible combinations of vertex mappings and the resulting mappings for their parents. The table assumes that a vertex in the first ontology has two children that are mapped to the children of a vertex in the second ontology. Column 1 in the table shows the mapping type of the first child

and column 2 shows the mapping type of the second child. Column 3 shows the deduced mapping type between the two vertices in the *Fully Mapped (FM)* scenario and column 4 shows the deduced mapping type between the two parents in the *Partially Mapped (PM)* scenario. A *User-defined* entry in the table indicates that the parent’s mapping type cannot be automatically deduced and the user has to provide the appropriate mapping type by hand. These deduction operations can easily scale up to include the cases where a vertex has more than two children. They will be performed recursively, starting from the vertices that are aligned and traveling up the global ontological tree, to deduce the mapping types between the ontology and the local ontologies. As previously mentioned, all combination results can be overridden by the user.

Table 2. Automatic mapping deduction operations

Child 1	Child 2	FM	PM
Exact	Exact	Exact	Subset
Exact	Approximate	Approximate	Subset
Exact	Superset	Superset	User-defined
Exact	Subset	Subset	Subset
Exact	Null	Superset	User-defined
Approximate	Approximate	Approximate	Subset
Approximate	Superset	Superset	User-defined
Approximate	Subset	Subset	Subset
Approximate	Null	Superset	User-defined
Superset	Superset	Superset	User-defined
Superset	Subset	User-defined	User-defined
Superset	Null	Superset	User-defined
Subset	Subset	Subset	User-defined
Subset	Null	User-defined	User-defined
Null	Null	User-defined	User-defined

4.3 Ontology Merging

Each local ontology might have a different organization of the entities based on the primary function of the agency maintaining it. For example, a county in which agriculture is the main occupation will have more categories of agricultural land usage than the global ontology drawn up for the state. When such a local ontology is aligned to the global ontology, there might be several places where the mapping type is *Null* or the user has to provide mappings because the automatic alignment process fails. This can indicate that a particular criterion of classification is missing in the global ontology and leads to loss in the resolution of the data when local ontologies using that classification technique are aligned. In such cases, the expert in charge of maintaining the global ontology could add the missing classification, that is, could *merge* concepts from a local ontology into the global ontology. This can be viewed as merging concepts from local ontologies into the global ontology.

In the global ontology of Figure 5, commercial land usage is classified as *Commercial Sales* and *Commercial Service* (based on the primary function of the commercial establishment). In the local ontology, commercial land usage is sub-classified as *Commercial Intensive* and *Commercial Non-intensive* (based on the size of the operations). The two parent vertices are considered aligned, because their level of detail is similar. As shown in Figure 5, vertices *Commercial Sales* and *Commercial Service* cannot be mapped to any of the vertices in the local ontology and hence have their mapping type as *Null*. Therefore, the mapping type between *Commerce* and *Commercial Sector* cannot be automatically deduced and is specified as *Approximate* by the user. This mapping type denotes that the mapping between two vertices seems right, but the subclassification is not along the same characteristics.

Classification of commercial land usage, based on the scale of operations, is missing from the ontology and could be introduced to better align local ontologies using that classification scheme. The alignment of the ontologies after the additional level of classification was introduced is shown in Figure 6.

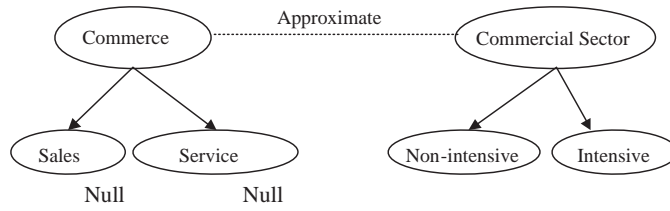


Fig. 5. Ontology alignment before the deduction process

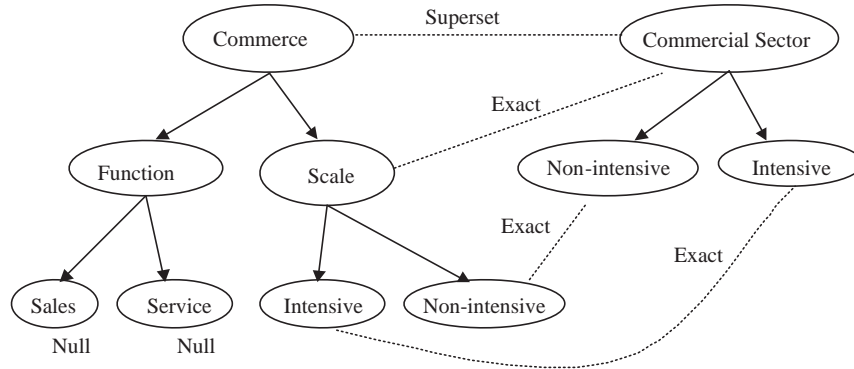


Fig. 6. Ontology alignment after deduction

5 Agreement Maker

The *Agreement Maker* is a software tool that is used to create the mappings between the global ontology and a local ontology and generate an agreement document, which is used by the query processor. The query processor maps a query expressed in the terms used in the global ontology to the local ontologies [3].

The local expert maps the ontology of the local database to the global ontology with a user interface, which shows the two hierarchies that represent the ontologies in two separate panes, allowing the expert to browse through the contents of each of the ontologies and establish the mappings between an entity (or entities) in the global ontology and an entity (or entities) in the local ontology. Figure 7 shows that interface.

5.1 Agreement Document

The local expert selects semantically related entities in both ontologies and chooses one of the mapping types from Section 4.2, which would best explain the relation between the concepts. There is a button labeled 'Update Mapping' that saves the mapping into a table. Each row in the table shows the names of the two vertices that are mapped (one from the global ontology and the other from the local ontology), and the type of mapping between them. As soon as the mapping is saved, all the vertices that have been mapped get clearly marked (with '→', as shown in Figure 7). In the backend, the agreement document also gets updated.

5.2 Deduction Module

The semi-automatic alignment methodology from Section 4.2 is integrated into the Agreement Maker system to simplify the task of aligning large ontologies. Once the children of two vertices (one belonging to the global ontology and the other one to the local ontology) are mapped by the user, the mapping between the parents of those vertices can be inferred by the automatic mapping deduction operations of Table 2. These results

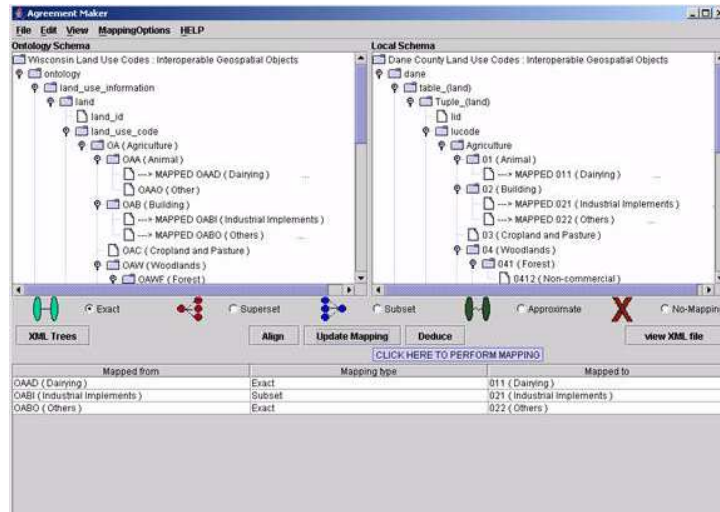


Fig. 7. User interface showing established mappings

are then presented to the user, who may choose to integrate them in the agreement document or discard them.

However, not only mappings that result from the deduction operations are added to the agreement document. Mappings can also result from the comparison of the labels of vertices, which have either the same name (e.g., *person_name*), which will be mapped using *Exact*, or because by using a vocabulary it is found that two terms are synonyms, or because the comparison of the respective strings (e.g., using an edit distance algorithm) produces a high score. Such mappings also reduce the load on the user. A variety of such algorithms for comparison of terms is available. Of course, the user’s decision overrides any mappings by the automatic deduction mechanism. Also, if the user has already mapped a vertex, no deductions will be considered for that vertex.

The results obtained by using the automatic deduction module are displayed in a separate pop-up window, which is shown in Figure 8. The tabular results indicate the vertices of the global ontology that are mapped to vertices of local ontology, and the respective mapping type. It also indicates whether the mapping was performed by the user, by the deduction mechanism, or by the tool box (which we named “Magic Box”) for the comparison of the labels of the vertices. The user can either commit the changes to the Agreement File if the results seem appropriate or discard them. If the changes are committed, they will be displayed in the main user interface.

6 Discussion

In Section 4 we present the assumptions we have made for the semi-automatic alignment to work. One of them is that the specialization must be total, otherwise the automatic deduction operation may produce semantically incorrect results. Next we consider two cases: in the first case, the specialization of the global ontology is not total and in the second case, the specialization of the local ontology is not total. For the sake of our discussion, we consider a simple example based on residential parcels where we assume that the only possible specializations of *Residential Buildings* are *One-family residence* and *Two-family residence*.

Case 1 In the first case we consider two subtrees of the local and of the global ontologies, where all the children of *Residential Buildings* and all the children of *Apartment Buildings* are represented in Figure 9. In Figure 9 the user initially maps *Two-family residence* to *Multiple-family residence* using *Subset* as the mapping type and maps *One-family residence* in the global ontology subtree to *One-family residence* in the local ontology subtree using *Exact* as the mapping type. Running the deduction process of Table 2, the parent vertex of the global ontology subtree *Residential Buildings* is mapped as *Subset* of the parent of the local

Mapped from	Mapping type	Mapped to	Mapped by
Wisconsin Land Use Codes: Interop	Exact	Dane County Land Use Codes: Interop	Deduction Table
ontology	Exact	dane	Deduction Table
land_use_information	Exact	table_(land)	Deduction Table
land	Exact	tuple_(land)	Deduction Table
land_id	Exact	id	Magic Box
land_use_code	Exact	lucode	Magic Box
QA (Agriculture)			
QAA (Animal)	Superset	01 (Animal)	Deduction Table
QAAD (Dairying)	Exact	011 (Dairying)	User
QAOD (Other)			
QAB (Building)	Subset	02 (Building)	Deduction Table
QABI (Industrial Implements)	Subset	021 (Industrial Implements)	User
QABO (Others)	Exact	022 (Others)	User
QAC (Cropland and Pasture)			
QAW (Woodlands)			
QAWF (Forest)			
QAWFC (Commercial)			
QAWFN (Non-commercial)			
QAWN (Non-forest)			
QAH (Housing)			
QAHF (Farm Residence)			
QAHO (Other)			
QAV (Vacant Land)			
QAVA (Farmland)			
QAVN (Non-farmland)			
QAT (Water Supplies)			

Fig. 8. The deduction result window

ontology subtree *Apartment Buildings*. This result is semantically incorrect: *Apartment Buildings* should be a subset of *Residential Buildings*, not the opposite. The system may notice a problem when the user will override the automatic alignment module to make *Residential Buildings* a *Superset* of *Apartment Buildings*. It is clear in this case that the specialization of *Residential Buildings* in the global ontology is not total: either there should be other children of *Residential Buildings* that are siblings of *One-family residence* and *Two-family residence* (e.g., *Three-family residence*, *Multiple-family residence*) or there should be a vertex that generalizes *One-family residence* and *Two-family residence*, is a child of *Residential Buildings*, and is a subset of *Apartment Buildings* in the local ontology. Note that ideally one would want the global ontology to be as “complete” as possible, but, as mentioned before, sometimes the local domain might contain more detail on certain concepts, since it specializes in a certain domain.

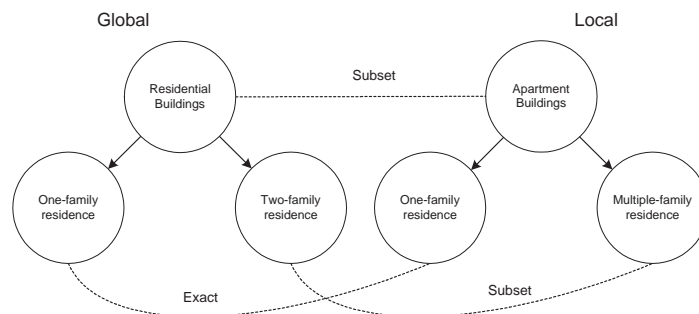


Fig. 9. Case 1: “Incomplete” global ontology

Case 2 In this case, we interchange the ontologies of the previous case as shown in Figure 10. This problem illustrates the case where the local ontology is less “complete” than the global ontology. This may be a rather common case, where the local ontology models the reality of a situation that is more limited. The local ontology may represent a town district where a *Residential Building* can only be a *One-family residence* or a *Two-family residence* and therefore these are all the representatives of *Residential Buildings*. Again, the deduction mechanism captures the structural aspect but not the semantic one, and erroneously concludes that *Apartment Buildings* is a superset of *Residential Buildings*.

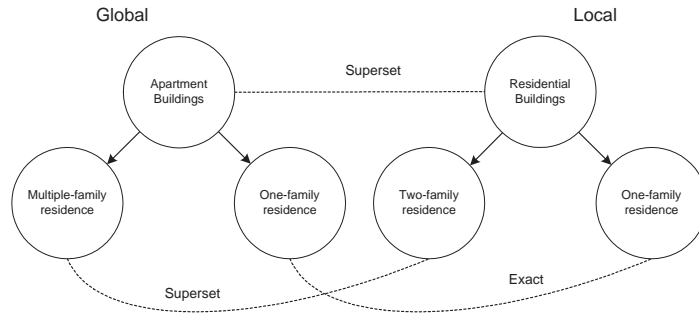


Fig. 10. Case 2: “Incomplete” local ontology

Proposed Solution To analyze the situation we consider Figure 11, where we represent “monotonic” mappings between two different trees. For example, vertex B maps as *Exact* to vertex G . Since vertex G represents a *Superset* of vertex H and a *Subset* of vertex F , then vertex B is a *Superset* of vertex H and a *Subset* of vertex F . In order to guarantee monotonicity, the situation where we have a “bowtie” cannot exist (recall that it was our assumption in Section 4 that “bowties” were not allowed and we will not be relaxing that assumption now).

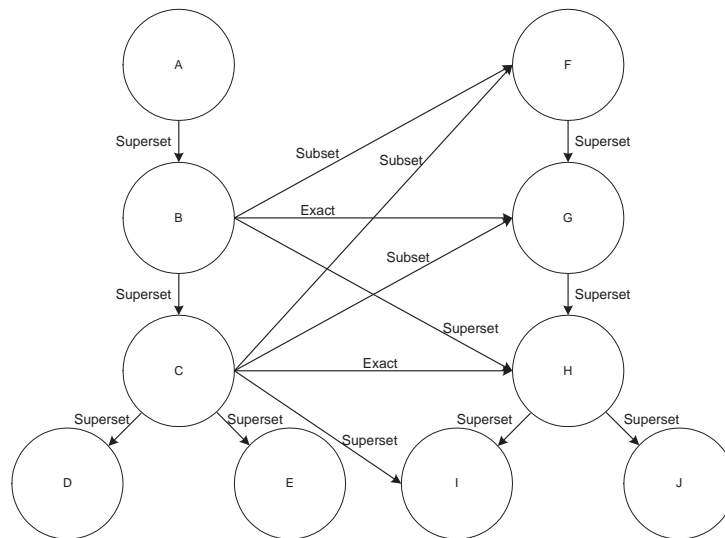


Fig. 11. Monotonic mappings

A possible solution for the first case is to augment the global ontology so as to encompass the local ontology. When the user maps *Residential Buildings* as a *Superset* of *Apartment Buildings* the system should catch such inconsistent mapping with the one produced by the automatic deduction mechanism and suggest the introduction of a virtual vertex (see Figure 12). A virtual intermediate vertex *Apartment Buildings* is inserted in the global ontology that carries the same name of its equivalent in the local ontology. The mapping remains *Subset* because structurally what *Apartment Buildings* contains in the global ontology is a *Subset* of what *Apartment Buildings* contains in the local one. Because a set is both a subset and a superset of itself the deduction mechanism is correct (but not as tight as it could be; in this case, the string comparison mechanism that we have mentioned in Section 5 would produce the correct *Exact* mapping). As for *Residential Buildings* in the global ontology, it can possibly be mapped to the immediate parent of

Apartment Buildings in the local ontology or an exact match could be found further up in the local ontology. The introduction of *Apartment Buildings* in the global ontology will improve the granularity of the queries on the local ontology, as the user can now specifically query for “Apartment Buildings” and obtain a precise response from the local ontology. Likewise, more children can be added to the vertex *Residential Buildings* to encompass other local ontologies.

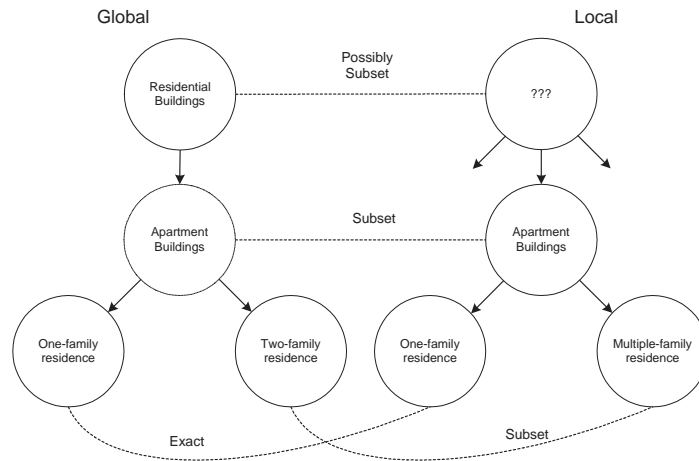


Fig. 12. Insertion of a virtual vertex

As for the second case, and taking into consideration the monotonic characteristic of the mappings, an ancestor of *Apartment Buildings* in the global ontology will be more likely to yield an exact match to the vertex *Residential Buildings* in the local ontology (Figure 13).

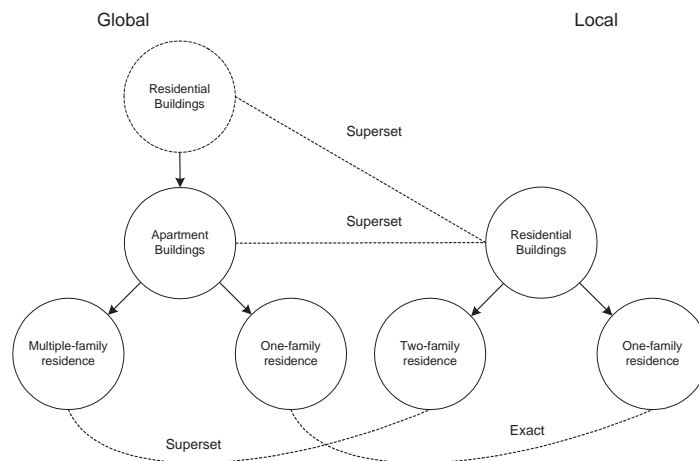


Fig. 13. Consideration of an ancestor

Scoring functions [4] can also capture the essence of the monotonicity of the trees. For example the value of a scoring function would improve until we find a more suitable vertex that is a parent of *Residential Buildings*. After an exact match is found, the scoring function would deteriorate for the parents of the vertex that yielded an exact match.

7 Conclusions

In this paper, we describe an ontology-driven approach to data integration and how we use such an approach in the geospatial domain. Data is integrated based on a global ontology of land usage patterns. The end user poses queries in terms of entities in the global ontology, which are then executed against the local ontologies. To enable query processing, the different ontologies have to be aligned. The relationships between aligned entities in the ontology and in each data source are expressed using different types of mappings, which we have defined so as to capture the range of possible semantics in our domain.

To allow for the semi-automatic processing of the ontology alignment, we have proposed a framework that defines the values associated with the vertices of the global ontology in two possible ways: as functions of the values of the children vertices or of the user input. We discussed some of the problems that we would encounter by relaxing one of our simplifying assumptions and possible solutions to those problems. We are currently conducting experiments on aligning ontologies to test the effectiveness of our semi-automatic processing.

There are several related issues that still need to be investigated, pertaining to the alignment process. In particular, more applications need to be investigated to possibly find other mapping types and to consider different kinds of ontologies. While our mapping types are “adequate”, to express the mappings in our current application, we would like to characterize the notion of *adequacy* for such mappings, especially as integrated with the alignment process. Another issue we are trying to resolve is that of vertices that intersect in meaning.

We have made some simplifying assumptions concerning the alignment process, which can be rather complex. Our simplifying assumptions may or may not hold in other applications or for other types of ontologies. Therefore, we plan to continue the work we started in Section 6.

Acknowledgments

We would like to thank Nancy Wiegand and Steve Ventura, from the Land Information & Computer Graphics Facility at the University of Wisconsin-Madison, for the discussions on land use problems. We are also indebted to Naveen Ashish for pointing to us some interesting problems and to Afsheen Rajendran for the initial implementation of the prototype.

References

1. H. Benetti, D. Beneventano, S. Bergamaschi, F. Guerra, and M. Vincini. An Information Integration Framework for E-Commerce. *IEEE Intelligent Systems*, 17:18–25, 2002.
2. S. Bergamaschi, F. Guerra, and M. Vincini. A Data Integration Framework for E-Commerce Product Classification. In *1st International Semantic Web Conference (ISWC)*, pages 379–393, 2002.
3. I. F. Cruz, A. Rajendran, W. Sunna, and N. Wiegand. Handling Semantic Heterogeneities using Declarative Agreements. In *International ACM GIS Symposium*, pages 168–174, 2002.
4. E. Hovy. Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses. In A. Rubio, N. Gallardo, R. Castro, and A. Tejada, editors, *First International Conference on Languages Resources and Evaluation (LREC)*, Granada, Spain, pages 535–542. 1998.
5. D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. An Environment for Merging and Testing Large Ontologies. In *Seventeenth International Conference on Principles of Knowledge Representation and Reasoning (KR-2000)*, pages 483–493, 2000.
6. G. A. Miller. WordNet: An Online Lexical Database. Technical report, Princeton University, 1990.
7. N. F. Noy and M. A. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *The Sixteenth National Conference on Artificial Intelligence (AAAI)*, pages 450–455, 2000.
8. J. Park, J. Gennari, and M. Musen. Mappings for Reuse in Knowledge-based Systems. In *11th Workshop on Knowledge Acquisition, Modelling and Management, KAW 98*, 1998. <http://ksi.cpsc.ucalgary.ca/KAW/KAW98/park/>.
9. J. Sowa. Building, Sharing, and Merging Ontologies, 2001. <http://www.jfsowa.com/ontology/ontoshar.htm>.