

Handling Semantic Heterogeneities using Declarative Agreements

Isabel F. Cruz*, Afsheen Rajendran, and William Sunna
Department of Computer Science
University of Illinois at Chicago

Abstract

The focus of this paper is on interoperability issues to achieve data integration in distributed databases in a geospatial application. Our concrete application is in the context of the State of Wisconsin Land Information System (WLIS). In WLIS, data is stored in XML using independently maintained local databases. However, answers to many queries must span several databases. Our approach is based on the existence of a central ontology and on declarative transformations, called *agreements*, between the schemas of the local databases and the central ontology. Using our approach, end users can seamlessly query and aggregate semantically related data that is available throughout the state using a visual interface. An expert at the site of the local database uses another visual interface to specify the agreement between that database and the ontology. We have performed user studies for both interfaces.

Keywords: Geospatial interoperability, ontologies, user interfaces for GIS

1 Introduction

The number and accessibility of geographical data sources have increased tremendously due to the rise of the Internet and other distributed computing systems. Data gathered and maintained by independent agencies are being published on the Internet. Users access and analyze data from multiple sources, to derive new insights into problems. However, GISs were not initially designed to work in cooperation with other such systems and hence their integration poses several interoperability problems [9].

Overcoming interoperability problems by handling syntactic and semantic heterogeneities has been widely discussed [3, 12, 14]. Interoperability problems range from those occurring due to differences in network protocols and operating systems to those occurring due to differences in underlying assumptions and semantics. One of the challenges is to handle interoperability problems at the semantic level and relieve the end user of the need to have prior knowledge of the underlying syntax and semantics of the different data sources [4, 17].

To achieve interoperability, a system should understand the semantics of the user's query and use the available resources as well as it can to provide meaningful answers. Such a system can then support high-level, context-sensitive queries over multiple data sources, irrespective of the underlying heterogeneities [17].

Several solutions to data integration have been proposed, ranging from federated databases with schema integration to mediation approaches using ontologies. Ontologies can be used to drive all aspects and components of an information system, leading to ontology-driven information systems [11, 13]. Such systems are particularly helpful for solving GIS interoperability problems that arise due to the heterogeneity of geographical data.

*Corresponding author. Email address: ifc@cs.uic.edu

This paper proposes a solution for overcoming the semantic interoperability problems involved in integrating data between heterogeneous databases by using declarative encodings of relations between the schema that describes the overall domain, henceforth called the *ontology*, and the schemas of the local databases. To illustrate the approach, an interoperability problem in a land use information system being developed for the state of Wisconsin is identified and a solution is proposed.

1.1 Wisconsin Land Information System

The Wisconsin Land Information System (WLIS) [18] is being planned as a distributed Web-based system with heterogeneous data residing on local and state servers. One of the most important functions for WLIS is to integrate land use data across jurisdictions to enable decision-making for comprehensive land use planning. XML [5] will be used to represent WLIS land use data to facilitate processing by emerging XML search and query engines and to serve as the common ground between different database technologies. However, heterogeneity in the data sources requires semantic integration before a goal of full query support can be met.

A common reason for the semantic heterogeneities is the usage of different land use coding conventions in different databases. Land use data are often found in parcel databases, and currently each jurisdiction has its own land use coding system. This results in a category of land usage having different land use codes in different databases. In some cases, the exact equivalent of a land use code in one database might be a collection of land use codes in another database. Moreover, the name of the attribute might not be “land_use_code”, but a locally used name like “lucode” or even “tag”. To be able to plan across jurisdictional boundaries, the diverse land use data sets need to be integrated, by resolving the different land use codes. These problems are discussed in detail in [19].

This problem of mapping between the different names and values of the attribute “land_use_code” is found usually in the case of attributes with pre-defined values, which are arranged in some hierarchical fashion. This paper suggests a solution to these problems by using *agreements*. An agreement is an XML document that declaratively represents the mapping of the entities in the local database database to the corresponding entities in the ontology. Each database has a corresponding agreement, which holds the local contextual information necessary for accurate mapping.

An end-user query is posed in terms of the entities in the ontology and is executed on one or more of the local databases. The agreement file is used for converting the query into sub-queries posed in terms of the entities in the local database. A sub-query is in the form of an XPath [7] expression and can be executed on the corresponding local database, which is represented as an XML graph using the framework of classes proposed in [6]. The results from all local databases are then gathered by the query processor and displayed to the user in terms of the ontological entities.

The paper is organized as follows. Section 2 illustrates the problems arising due to the use of different land use codes with an example. The system architecture and working are explained in the sections 3 and 4 respectively. The user interface components developed for the system are described in Section 5. Some related work is discussed in section 6. The last section brings up the conclusions and future work.

2 Land Use Codes

Two common examples of land use classification schemes: exhaustive lists and hierarchical models are shown in Table 1. Structurally these two systems are quite different, with the hierarchical model being more highly organized than the exhaustive list.

Now consider a typical query such as, “Where are all the crop and pasture lands in Dane County?”. A query of this kind is relatively straightforward when using one data set but more difficult when posed over a larger geographic area. Table 2 illustrates the heterogeneity of attribute names and values that would satisfy the query’s criteria over selected multiple data sets.

“Lucode”, “Tag”, “Lu1” and “Lu_4_4” must be resolved as synonyms for the “land_use_code attribute” in the ontology. Another issue that must be resolved is that the Racine County database has two land use codes for denoting croplands, while other counties have only one.

The mapping between the attribute name in the ontology and that in the local schema is usually “one-to-one”. The resolution of cases where the mapping is not “one-to-one” is an area for future research.

The equivalent of an attribute value in the ontology, can be a single value or a collection of values in the local schema. An example for the latter case, found in the Eau Claire County land use coding system is given in Table 3. In some cases there might not be an equivalent value.

The equivalent of the land use code “Commercial Intensive” in the ontology is the hierarchical collection of the codes “2.1.1” and “2.1.2”, while that of the land use code “Sales” is the non-hierarchical collection of the codes “2.1.1” and “2.2.1”.

3 System Architecture

The architecture of the system is shown in Figure 1. It is a modified version of the overall architecture proposed in [16]. A brief description of each component in our approach is given below.

1. GIS

This component encapsulates the local GIS database and the corresponding agreement file, which serves as the wrapper over the underlying data. The local database is represented as an XML document. The component contains classes written in Java, to treat the local database as a graph and to execute XPath expressions on it using the framework of classes proposed in [6]. As the relevant contextual information is stored declaratively in the agreement, the classes can be reused for other local databases without any modifications.

2. Ontology

This component encapsulates the ontology or the common knowledge necessary for meaningful interactions with the local databases. The end user query is posed in terms of the entities in the ontology. So this serves as the logical database schema to which all local databases conform to.

3. Semantic Registry

This component serves as the registry of all available local databases in the system. Whenever

	<i>Exhaustive Model</i>		<i>Hierarchical Model</i>
009	Shopping Center	1	Urban and Developed Land
010	Open Water	1.01	Residential
111	Single Family	1.01.01	Single Family Detached or Duplex
113	Two Family	1.01.02	Mobile Homes Not in Parks
116	Farm Unit	1.01.03	Multi-family Dwellings
140	Mobile Home	1.01.03.01	Three Unit Multi-family
		1.01.03.02	Four Unit Multi-family

Table 1: Different land-use classification schemes.

<i>Planning Authority</i>	<i>Attribute</i>	<i>LUCode</i>	<i>Description</i>
Dane County RPC	Lucode	91	Cropland Pasture
Racine County (SEWRPC)	Tag	811	Cropland Pasture and
		815	Other Agriculture
Eau Claire County	Lu1	AA	General Agriculture
City of Madison	Lu_4.4	8110	Farms

Table 2: Heterogeneity of attribute names and values.

2 Commercial
2.1 Commercial - Intensive
2.1.1 Sales
2.1.2 Service
2.2 Commercial - Non-intensive
2.2.1 Sales
2.2.2 Service

Table 3: Commercial land use classification in Eau Claire county.

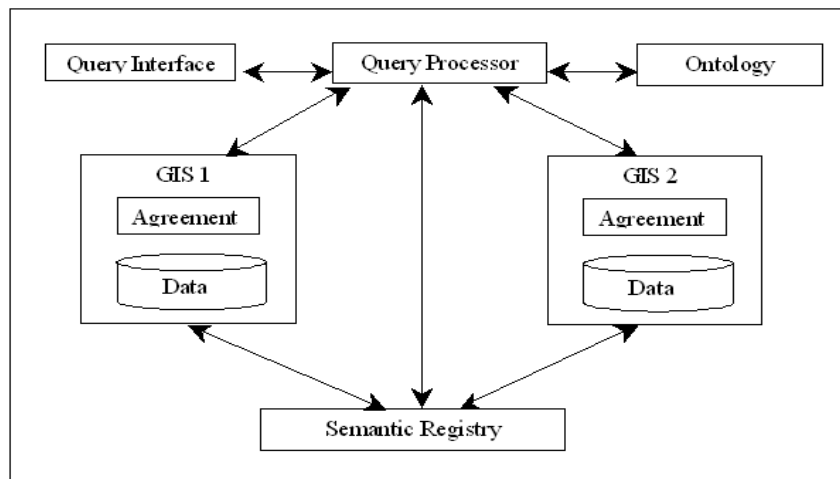


Figure 1: System architecture.

a new local database is added to the system, the registry is to be updated with the corresponding agreement file. The query processor component uses this component to execute a sub-query on a local database.

4. Query Processor

This component gets the query posed by the user using the query interface component and re-writes it into one or more sub-queries after consulting with the ontology and the agreement files of the local databases. The sub-queries are then executed on each local database and the results are gathered and shown to the user using the query interface component.

5. User Interface

There are two user interface components in the system. One is meant to be used by the end user to execute queries against the land use databases. The other is to be used by the

local expert to create the agreement file for the local database. They are discussed further in Section 5.

4 System Implementation

4.1 Land Use Data

Sample data was created to represent a land tax system that stores information about land parcels and their owners. Usually such a system is implemented in a file-based or a relational database, as described in [1]. In our approach, the data is in the form of XML documents. This facilitates processing by emerging XML search and query engines and serves as the common ground between different database technologies. Also, this is the format that will be used by WLIS. XML capabilities are being provided currently by most commercial database vendors. Actual land parcel data for Dane County in Wisconsin are stored in digital format and can be accessed at [8].

The two major entities, in such a system are “land” and “owner”. Sample data about a land parcel and its owner is given in Figure 2. Most land parcel data that we have encountered is quite similar to this particular example. 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 geometric attributes “l”, “r”, “t”, and “b”) and information about the owner of the parcel (“owner.id”). Owner data usually contains the name “name”), date of birth (“dob”), social security number (“ssn”), and gender (“gender”) of the owner.

```
<farm>
  <lid> lid21 </lid>
  <lucode> OA </lucode>
  <l> 100 </l>
  <r> 150 </r>
  <t> 100 </t>
  <b> b21 </b>
  <owner_id> 124-45-5678 </owner_id>
</farm>
<owner>
  <name> John Doe </name>
  <dob> 12-30-1977 </dob>
  <ssn> 124-45-5678 </ssn>
  <gender> M </gender>
</owner>
```

Figure 2: Sample land use data.

4.2 Ontology

The ontology acts as a single logical database schema and enables uniform querying across the local databases. An expert, who is familiar with the domain entities and relationships, can develop the ontology. A part of the *ontology.xml* document, which is the XML representation of the ontology, is shown in Figure 3.

```

<database id="ontology">
...
<table id="land_use_information">
<tuple id="land">
  <attrname id="land_id"/>
  <attrname id="land_use_code">
    <attrvalue id="OA" exp="Agriculture">
      <attrvalue id="OAG" exp="Agriculture - General">
        <attrvalue id="OAGC" exp="Agriculture - General - Cropland"/>
        <attrvalue id="OAGA" exp="Agriculture - General - Animal Husbandry"/>
      </attrvalue>
      <attrvalue id="OAW" exp="Agriculture - Woodlands">
        <attrvalue id="OAWF" exp="Agriculture - Woodlands - Forest">
          <attrvalue id="OAWFC" exp="Agriculture - Woodlands - Forest - Commercial"/>
          <attrvalue id="OAWFN" exp="Agriculture - Woodlands - Forest - Non-commercial"/>
        </attrvalue>
        <attrvalue id="OAWN" exp="Agriculture - Woodlands - Non-forest"/>
      </attrvalue>
      <attrvalue id="OAH" exp="Agriculture - Housing"/>
    ...
    <attrvalue id="OAO" exp="Agriculture - Others"/>
  </attrvalue>
</attrname>
  <attrname id="shape_file"/>
  <attrname id="owner_id"/>
</tuple>
</table>
...
</database>

```

Figure 3: Ontology represented as an XML document.

Each database might have a different organization of the attribute values based on the primary function of the agency maintaining it. In some mediation approaches, when local database schemas are mapped to a central schema, data at lower levels in the local database are aggregated into data at higher levels. Resolution of data is sacrificed in order to ensure conformance. In our approach, there is a facility for storing the different types of classification of data in the ontology to enable local databases to be mapped without loss in data resolution.

For example, commercial land usage can be classified as “commercial sales” and “commercial service” (based on the function) in one database and as “commercial intensive” and “commercial non intensive” (based on the size of operations) in another database. If only one type of classification is supported in the ontology, data in the other database has to be aggregated into a single category, for example ‘commercial’. As both the types of classifying commercial land usage can be specified in the ontology, it is possible to preserve the resolution of data available in the local databases in our approach.

4.3 Agreements

The agreement file contains the mappings between entities in the local database to the ontology. It is to be created whenever a local database is to be added to the system. A local expert, familiar with the semantics of the database entities and how they relate to the entities in the ontology, can develop the agreement file.

The mapping between entities like *database*, *table*, *tuple*, and *attribute* in the ontology and that in the local schema is usually direct. The local equivalent of each such entity is specified in the “equiv” attribute of the tag representing the entity.

These mappings are written in the agreement file as

```
<table id="land_use_information" mapping="one-to-one" equiv="tbl_land">
<attrname id="land_id" mapping="one-to-one" equiv="lid"/>
```

On the other hand, an attribute value in the ontology can be mapped to a single value or a collection of attribute values in the local database. In some cases, multiple attribute values in the ontology can map to the same local attribute value. The mappings are broadly classified as:

1. *one-to-one*

In this type of mapping, there is a direct relationship between the attribute value in the ontology to that in the local schema. For example, the land use code “OAC” (Agriculture - Croplands and Pasture) can be directly mapped to the land use code “91” (Cropland and Pasture) in the Dane County database.

This mapping is written in the agreement file as

```
<attrvalue id="OAC" mapping="one-to-one" equiv="91"/>
```

2. *one-to-many*

In this type of mapping, an attribute value in the ontology is equivalent to the collection of attribute values in the local schema. For example, the land use code “OIO” (Industrial - Others) can be mapped to the land use codes “35” (Scientific Instruments) and “36” (Miscellaneous Industrial) in the Dane county database.

This mapping is written in the agreement file as

```
<attrvalue id="OIO" mapping="one-to-many">
  <localvalue> 35 </localvalue>
  <localvalue> 36 </localvalue>
</attrvalue>
```

3. *many-to-one*

In this type of mapping, an attribute value in the local schema is equivalent to many attribute values in the ontology. For example, both the land use codes “OIML” (Industrial-Manufacturing-Plastics) and “OIMR” (Industrial - Manufacturing - Rubber) can be mapped to the land use code “31” (Manufacturing - Plastics and Rubber) in the Dane county database.

This mapping is written in the agreement file as

```
<attrvalue id="OIML" mapping="many-to-one" equiv="31"/>
<attrvalue id="OIMR" mapping="many-to-one" equiv="31"/>
```

4. *one-to-null*

An attribute value in the ontology might not have an equivalent value in the local schema. For example, the land use code “OAWN” (Agriculture - Woodlands - Non-forest) has no equivalent value in the Dane county database. In such cases, the local database expert might want to specify another land use code to be used instead. This can be specified in the “bettercode” attribute in the agreement file. As more analysis is needed for specifying this attribute, such changes can be made in the agreement file after it is generated using the interface described in Section 5.2.

This mapping is written in the agreement file as

```
<attrvalue id="OAWN" mapping="one-to-null" bettercode="OAW">
```

5. *children*

In this type of mapping, an attribute value in the ontology is equivalent to the collection of attribute values in the ontology itself. The constituent attribute values are in turn mapped to those in the local schema, by any one of the five mappings.

For example, the land use code “OAWF” (Agriculture - Woodlands - Forests) is the collection of the land use codes “OAWFC” (Agriculture - Woodlands - Forests - Commercial) and “OAWFN” (Agriculture - Woodlands - Forest - Noncommercial). The land use codes “OAWFC” (Agriculture - Woodlands - Forests - Commercial) and “OAWFN” (Agriculture - Woodlands - Forest - Noncommercial) are mapped “one-to-one” to the land use codes “94” and “99” respectively.

This type of mapping depends only on the hierarchical ordering of the attribute values in the ontology and cannot be specified by the user when the agreement file is generated using the agreement maker interface described in Section 5.2.

This mapping is written in the agreement file as

```
<attrvalue id="OAWF" mapping="children">
  <attrvalue id="OAWFC" mapping="one-to-one" equiv="94"/>
  <attrvalue id="OAWFN" mapping="one-to-one" equiv="99"/>
</attrvalue>
```

The basic ontology can be created based on the analysis of a small number of local databases. Then the agreement for each database to be integrated can be created by the local authority, in charge of maintaining the database. A visual user interface was developed to simplify the agreement creation process, which is described in Section 5.2.

4.4 Query Execution

Initially the list of available databases are shown to the user. The user selects the databases to be queried and then specifies the query in terms of the attribute names and values in the ontology. Sub-queries are generated by rewriting these queries in terms of the attribute names and values in the local schema. The sub-queries are then executed on the corresponding database. The results from all the selected databases are collected and then displayed to the user.

When the database file represented in XML is loaded using a DOM compliant parser, a DOM tree is returned. The XPath [7] engine in the Xalan-Java API provides a simple way of expressing

a path through a document tree to select a set of nodes. When a path expression is evaluated, the XPath facilities select a set of nodes relative to a context node. Therefore, executing a sub-query involves generating the corresponding XPath expression and executing it on the DOM tree. The user interface and query processing components were developed using Java. For this purpose, we use a slightly modified version of the API in [6].

5 User Interface Components

Two user interface components were created for the system. One is to be used by the end user to execute queries against the land use databases. The other is to be used by the local expert to create the agreement file for the local database.

5.1 Query Interface

Typically the queries posed by the end user involve selecting all the land parcels within a given area, which are classified under a particular land use category. To make this query interface intuitive, an extension to ArcView GIS known as Axiomap [10], was used to display interactive maps in a web browser.

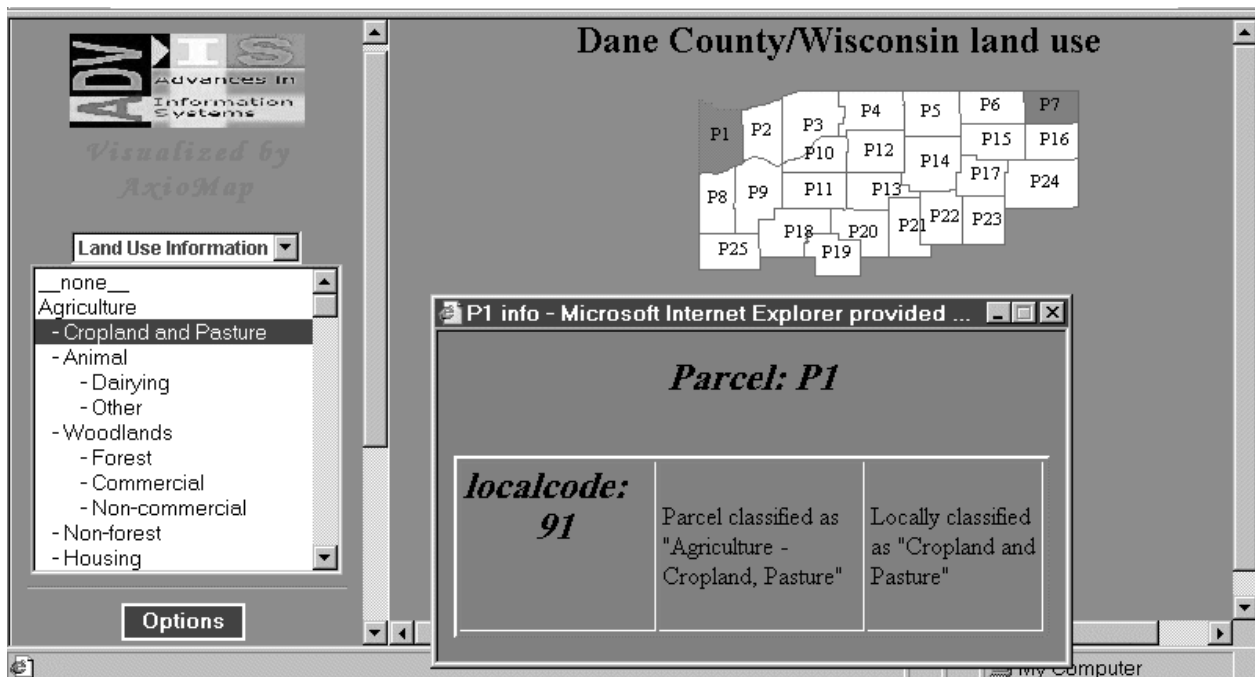


Figure 4: End-user query interface using interactive maps.

In Figure 4, a small region of Dane county is displayed showing the various parcels (a sample map of Dane county is used). The different land use codes are listed in the left frame. When the user selects a land use code in the left frame, all land parcels in the given region, which fall under that land use classification are highlighted in the map. For example, in Figure 4 the user is interested in finding out all parcels whose land use code is "Agriculture, Cropland and Pasture". Parcels P1 and P7, which fall under that land use classification, are shown highlighted on the map.

The user can learn more about a particular land parcel by selecting it to open a pop-up window to display additional information, as shown in Figure 4.

We performed a user study on this interface, in which seventeen people participated. The users liked the intuitive interface and the distribution of objects on the page with the use of different colors. The map layout and the additional information display were points of strength according to the study. However, the users had some problems with the loading time of the page especially when connected to the Internet using dial up modems. The users also reported quick resource consumption and some zooming and panning problems. Overall about 94 percent of those who tested the interface recommend it for other similar projects and gave it a rating of 7.3 on a scale of 10.

5.2 Agreement Maker Interface

The expert in charge of the local database uses the user interface shown in Figure 5, to create the agreement file. The ontology and the local database schema are shown in the top left and right panes respectively. The current set of mappings are shown in the bottom pane and help the user in keeping track of entities which have already been mapped and those which are yet to be mapped. The mapping options are shown in the center of the application window and can be chosen while specifying mappings.

The user selects an entity or a collection of entities in the ontology schema, the equivalent entity or collection of entities in the local schema, one of the four possible mapping options (one-to-one, one-to-many, many-to-one and one-to-null) and then asks the system to update the mappings using the “Update Mappings” button shown next to the mapping options. The user is not allowed to specify a mapping option as “children” because it depends only on the hierarchical ordering of the attribute values in the ontology and not on the user’s preference.

As the user specifies mappings for the entities in the ontology, the current set of mappings displayed in the bottom pane are updated to reflect the changes. The user can change the mapping of an entity in the ontology by selecting it and the equivalent entity or entities in the local schema and then specifying the new mapping option. Once all the entities in the ontology have been mapped, the user asks the system to create the agreement file using the “Write To File” button shown below the mapping options. After the agreement file is generated, the expert can make additional changes in it, if necessary. The user can invoke the help window using the “Help” button.

A preliminary user study was conducted for this interface, with five users. The users liked the ease of operation and the use of the feedback pane on the bottom. They found the help instructions very useful and layout of the objects on the screen attractive. The users also think that the running speed of the interface is very reasonable and they were able to understand and conduct all kinds of mappings with ease. However, 40 percent of the users believe that using such interface is not intuitive. This results contrast with very positive replies to the other eleven questions, including those on the appropriateness of system terminology, and adequacy of the presentation of the information. Clearly in a future study this question has to be rephrased and possibly split into different questions. Users also reported problems concerning an empty warning box that appears if the update button is clicked on but no nodes are selected. The users suggest adding an undo feature if mistakes were made and to color the strings in the feedback pane for better visibility. These suggestions have now been incorporated. Overall the interface was rated 8.6 on a scale of 10, and all the participants in the study recommended it for similar projects.

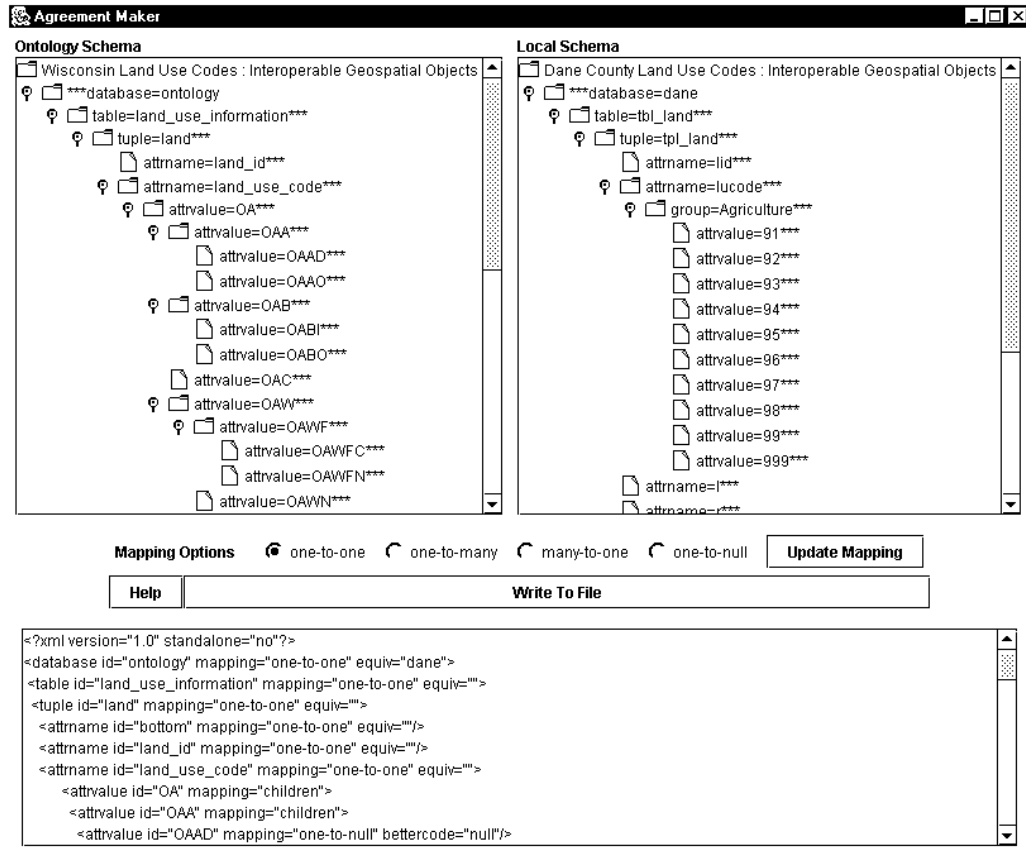


Figure 5: User interface for creating the agreement.

6 Related Work

Interoperability problems have been discussed in the literature since database integration studies started in the mid-80s. Interoperability problems occur at several levels. Differences can occur at the system level due to the usage of different operating systems and communication protocols; at the syntax and structure level due to the usage of different file formats, data typing mechanisms and interface definitions and at the semantic level due to different assumptions in data usage.

As the reach of computing systems has grown from within isolated organizations to across organizations and then across the Internet, the heterogeneities have increased and have led to further interoperability problems. To handle the heterogeneities, an ontology-driven approach has been suggested in [4, 17]. Ontologies are used to capture the knowledge in the application domain and use it to answer the information requests in the best possible way. In our approach, we use an ontology to capture the differences in schemas of the constituent databases and to serve as the format in which queries are posed by the user.

IGO (Interoperable Geospatial Object) [9], proposes a solution to the interoperability problem using a geospatial information interoperability envelope. The envelope contains metadata and usage characteristics in addition to the underlying geospatial data. Methods associated with the envelope can be invoked whenever the information is acted upon. In our approach, the envelope consists of a Java class which contains methods to access the local data and an agreement file to hold the relevant metadata.

ISIS (Interoperable Spatial Information System) [16] is a semantic mediation approach to GIS interoperability and proposes a multi-agent architecture and a multi-level data model. The multi-level data model is used to represent semantic contexts in addition to the traditional textual and spatial information, to provide a basis for the resolution of semantic differences and to transfer data between systems. The multi-agent architecture provides autonomy of both information sources and users, and flexible and dynamic interconnection of participating databases. Data transfer between the local and mediated schemas is achieved with the help of wrapper and co-operation agents. The wrapper agent is used to represent the local schema. The cooperation agent is used to convert queries expressed in terms of the mediator schema into local queries and return results. In our approach, the agreement file holds the schema of the local database and additional semantic information. The functions of the cooperation agent are performed by a Java class. The main difference is that in ISIS, the methods in the cooperation agent are procedural, while in our approach, due to the additional semantic information available in the agreement file, the data-access mechanisms are declarative. This enables new databases to be added to the system with minimal effort.

The approach in [6] used an application to visualize US election results to illustrate the problems in handling different XML representations. Though the election results were stored in XML, the structure of the documents varied from state to state. A framework of classes was developed to treat XML documents as graphs and collect data by executing XPath expressions against these graphs. A mediator schema was developed and lookup files were created for each state's XML representation to encode relations to the mediator schema. Due to the small number of hierarchical levels of data, it was possible to encode the XPath expressions in the lookup file. In our approach, the XPath expressions are created dynamically from the schematic and semantic information stored in the agreement in order to support arbitrary hierarchical levels.

7 Conclusions

Due to the increase in complexity and differences in representation of geospatial data sources, the process of integrating such data has met with interoperability problems. These problems can be solved using an ontology-driven approach to handle the semantic heterogeneities in the data sources. This paper proposes such a solution by using declarative encodings of relations between the entities in the ontology and those local databases. To illustrate the approach, an interoperability problem in a land use information system being developed for the state of Wisconsin was identified and a solution was proposed.

This work can be improved further by using advanced XML query languages like XQuery and XML-QL, which are more expressive than XPath, which was used in our current approach. The schemas used here were derived from relational databases. Furthermore, the ontology can be represented using F-Logic [15] to add inference mechanisms, which are helpful in the re-writing queries and optimizing their execution [2].

“One-to-one” mappings between tables, tuple and attributes were handled in this approach. The databases can be further analyzed and more complex mappings between such entities can be identified and ways to handle them can be devised.

8 Acknowledgements

We would like to thank Nancy Wiegand, Steve Ventura, and Dan Patterson for the discussions on land use problems and Chaitan Baru and Ilya Zaslavsky for allowing us to use the Axiomap system and answering our questions. We would also like to thank Sathish Kumar Murugesan for his

help in developing the user interface component for creating agreements and the query processing components.

This research was supported in part by the National Science Foundation under Award EIA-0091489 and CAREER Award IRI-9896052, and by the Advanced Research and Development Activity (ARDA) and the National Imagery and Mapping Agency (NIMA) under Award Number NMA201-01-1-2001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the National Imagery and Mapping Agency or the U.S. Government.

References

- [1] D. J. Abel, B. C. Ooi, K.-L. Tan, and S. H. Tan. Towards Integrated Geographical Information Processing. *International Journal of Geographical Information Science*, 12:353–371, 1998.
- [2] S. Bergamaschi, S. Castano, and M. Vincini. Semantic Integration of Semistructured and Structured Data Sources. *SIGMOD Record*, 28(1):54–59, 1999.
- [3] Y. Bishr. Semantic Aspect of Interoperable GIS. Technical report, Wageningen Agricultural University ITC, 1997.
- [4] Y. Bishr. Overcoming the Semantic and Other Barriers to GIS Interoperability. *International Journal of Geographical Information Science*, 12:299–314, 1998.
- [5] T. Bray, J. Paoli, and C. M. Sperberg-McQueen. Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation, 2000. <http://www.w3.org/TR/REC-xml>.
- [6] P. W. Calnan and I. F. Cruz. Object Interoperability for Geospatial Applications. In *Proceedings of the First Semantic Web Working Symposium*, pages 229–243, Stanford, CA, 2001.
- [7] J. Clark and S. DeRose. XML Path Language (XPath) Version 1.0. W3C Recommendation, 1999. <http://www.w3.org/TR/xpath>.
- [8] W. L. I. Clearinghouse. Clearinghouse Approaches to GIS Data Discovery, 2002. <http://wisclinc.state.wi.us/datadisc/orgmetabl1.html>.
- [9] J. Doughty, P. Jones, J. Nolan, and S. Hirsch. Interoperable Geospatial Objects. In *GeoComputation 99*, 1999.
- [10] Elzaresearch. Axiomap: Application of XML for Interactive Online Mapping, 2001. http://www.elzaresearch.com/landv/axiomap_ie5.html.
- [11] F. Fonseca and M. Egenhofer. Ontology-driven Geographic Information Systems. In *7th ACM Symposium on Advances in Geographic Information Systems*, 1999.
- [12] M. Gahegan. Characterizing the Semantic Content of Geographic Data, Models and Systems. *Interoperating Geographic Information Systems*, pages 71–84, 1999.
- [13] N. Guarino. *Formal Ontology in Information Systems*. IOS Press, 1998.
- [14] V. Kashyap and A. Sheth. Semantic Heterogeneity in Global Information System: The Role of Metadata, Context and Ontologies. In M. Papazoglou and G. Schlageter, editors, *Cooperative Information Systems: Current Trends and Directions*. London Academic Press, 1996.

- [15] M. Kifer, G. Lausen, and J. Wu. Logic Foundations of Object-Oriented and Frame-Based Languages. *JACM*, 42(4):741–843, July 1995.
- [16] E. Leclercq, D. Benlimane, and K. Yetongnon. ISIS: A Semantic Mediation Model and An Agent Based Architecture for GIS Interoperability. In *IDEAS*, 1999.
- [17] A. Sheth. Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics. In M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, editors, *Interoperating Geographic Information Systems*, pages 5–30. Kluwer Academic Press, Norwell, MA, USA, 1998.
- [18] N. Wiegand, I. F. Cruz, and S. J. Ventura. Digital Government: Integrating Metadata Development, XML, and DBMS Search and Query Techniques in a State of Wisconsin Land Information System, 2000. Technical Report - University of Wisconsin-Madison and University of Illinois at Chicago.
- [19] N. Wiegand, D. Patterson, N. Zhou, S. Ventura, and I. F. Cruz. Querying Heterogeneous Land Use Data: Problems and Potential. In *National Conference for Digital Government Research*, 2002.