Handling Semantic Heterogeneities Using Declarative Agreements

Isabel F. Cruz Afsheen Rajendran
William Sunna
Department of Computer Science
University of Illinois at Chicago
[ifc,arajendr,wsunna]@cs.uic.edu

Nancy Wiegand
University of Wisconsin-Madison
wiegand@cs.wisc.edu

ABSTRACT

The focus of this paper is on interoperability issues to achieve data integration in distributed databases for geographic applications. 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. Our approach has been fully developed and tested.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Spatial databases and GIS; H.2.5 [**Heterogeneous Databases**]: Data translation; H.5.2 [**User Interfaces**]: Graphical user interfaces (GUI)

Keywords

GIS interoperability, ontology, user interfaces for GIS

1. INTRODUCTION

As the web accessibility of geographic information systems (GISs) is increasing tremendously, there is the potential for users to access and analyze data from those systems. However, GISs were not initially designed to work in cooperation and hence their integration poses several interoperability problems [7].

Overcoming interoperability problems by handling syntactic and semantic heterogeneities has been widely discussed [9, 11]. 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 [13].

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 [8, 10]. Such systems are particularly helpful for solving GIS interoperability problems that arise due to the heterogeneity of geographic data.

This paper proposes a solution for overcoming the semantic interoperability problems involved in integrating data from heterogeneous databases by using declarative encodings of the 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.

The Wisconsin Land Information System (WLIS) 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 is 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 the 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 is often found in parcel databases, and currently each jurisdiction has its own land use coding system. This results in different land use codes in each database. These problems are discussed in detail in [14].

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. This paper proposes *agreements* to achieve data integration. An agreement is an XML document that declaratively represents the mapping of the entities in the local database to the corresponding entities in the ontology. Each database has an agreement file, which

GIS'02, November 8–9, 2002, McLean, Virginia, USA.

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 [6] expression and can be executed on the corresponding local database, which is represented as an XML graph. The problem we consider is such that the answer to an end-user query is the union of the answers to the sub-queries. The results are then displayed to the user in terms of the ontological entities.

The paper is organized as follows. Section 2 uses an example to illustrate the problems arising due to the use of different land use codes. The system architecture and implementation are explained in Sections 3 and 4, respectively. The user interface components developed for the system are described in Section 5. Related work is discussed in Section 6. The last section brings up the conclusions and future work.

2. LAND USE CODES

Tables 1 and 2 show two common models of land use classification schemes, the exhaustive and hierarchical models, which are structurally quite different.

	Exhaustive Model
009	Shopping Center
010	Open Water
111	Single Family
113	Two Family
116	Farm Unit
140	Mobile Home

Table 1: Exhaustive classification scheme.

	Hierarchical Model
1	Urban and Developed Land
1.01	Residential
1.01.01	Single Family Detached or Duplex
1.01.02	Mobile Homes Not in Parks
1.01.03	Multi-family Dwellings
1.01.03.01	Three Unit Multi-family
1.01.03.02	Four Unit Multi-family

Table 2: Hierarchical classification scheme.

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 3 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", which is the corresponding 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.

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

Table 3: Heterogeneity of attributes and values.

3. SYSTEM ARCHITECTURE

The architecture of our proposed system is shown in Figure 1. It is a modified version of the architecture proposed in [12]. It consists of the following components:

1. GIS
   This component encapsulates the local 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. As the contextual information that pertains to each database is encoded in the agreement, the classes can be reused for other local databases without any modifications.

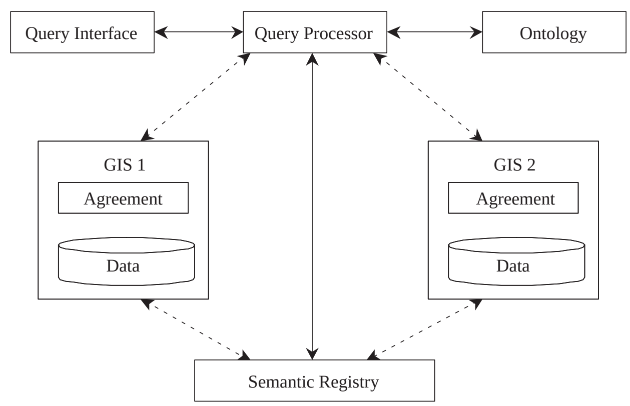

Figure 1: System architecture.

2. Ontology
   This component encapsulates the ontology, that is, the common knowledge necessary for meaningful interactions with the local databases. As the end user query is posed in terms of the entities in the ontology, the ontology is the logical database schema to which all local databases must map.

3. Semantic Registry
   This component serves as the registry of all available local databases in the system. Whenever a new local database is added to the system, the registry is 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 rewrites the query 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 used by the end user to express the queries on the land use databases. The other one is used by the local expert to create the agreement file for the local database.

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. Such a system can be implemented using a file-based or a relational database, as described in [1]. In our approach, the data is stored in the form of XML documents thus facilitating processing by emerging XML search and query engines and serving as the common ground between different database technologies. Actual land parcel data for Dane County in Wisconsin is stored in digital format at the Wisconsin Land Information Clearinghouse web site.

Two major entities are “land” and “owner”. Sample data about a land parcel and its owner is given in Figure 2. 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> 91 </lucode>
    <l> 100 </l>
    <r> 150 </r>
    <t> 100 </t>
    <b> 200 </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, develops the ontology. A fragment of the *ontology.xml* document, which is the XML representation of the ontology, is shown in Figure 3. The uniqueness of our approach is that precise classifications are specified in the ontology to enable local databases to be mapped without loss in data resolution. This is in contrast with some mediation approaches in which data at lower levels in the local database is aggregated into data at higher levels, when local database schemas are mapped to a central schema. That is, data resolution is sacrificed in order to ensure conformance.

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”. Using our approach, both types of commercial land usage can be specified in the ontology, therefore it is possible to preserve the data resolution available in the local databases.

4.3 Agreements

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

The mapping between the attribute name in the ontology and that in the local schema is usually “one-to-one”. That is, we consider the mapping between entities *database, table, tuple*, and *attribute* in the ontology and that in the local schema to be 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_info’’ 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. We classify the mappings as:

1. *one-to-one*
   There is a direct relationship between the attribute value in the ontology to that in the local schema. For example, the land use code “OAGC” (Agriculture - General - Croplands) 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*
   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:

```
<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 id="OAWO" exp="Agriculture - Woodlands - Others"/>
      </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.

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

3. *many-to-one*

   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. To help in querying for these codes, the local expert can either specify a code at a lower level of resolution, e.g. "OAW" (Agriculture - Woodlands) or group it along with a code at the same level of resolution, e.g. "OAWO" (Agriculture - Woodlands - Others). This decision is specified in the "bettercode" attribute in the agreement file. Such decisions can be made by the local expert when composing the agreement file with the user 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*

   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 - Non-commercial). The land use codes "OAWFC" and "OAWFN" are mapped "one-to-one" to the land use codes "94" (Commercial Forest) and "99" (Woodlands (Non-commercial Forest)), respectively. 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>
```

This type of mapping depends only on the hierarchical ordering of the attribute values in the ontology and cannot be specified by the local expert when composing the agreement file with the user interface described in Section 5.2.

4.4 Query Execution

Initially, the list of available databases is shown to the user. The user selects the databases to query 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 unioned 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 engine in the Apache 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 query processing components were developed in Java, using a modified version of the API in [4]. We note again that these components do not contain specific information related to the data being processed, which is solely contained in the agreement file. Therefore, they can be used without modifications in all local databases.

5. USER INTERFACE COMPONENTS

Two user interface components were fully developed and tested to serve as a concrete implementation of our approach. One is used by the end user to execute queries against the land use databases. The other one is used by the local expert to create the agreement file.

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. A standalone extension to ArcView GIS known as Axiomap,[1] was used to display interactive maps in a web browser.

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 all parcels with land use code "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.

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 is shown in the bottom pane and helps the user in keeping track of entities that have already been mapped and those that are yet to be mapped. The mapping options are shown in the center of the application window and can be chosen while specifying mappings.

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 thought that the running speed of the interface was 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 an interface is not intuitive. These results contrast with very positive replies to the other eleven questions, including those on the appropriateness of the 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. Overall the interface was rated 8.6 on a scale of 10, and all the participants in the study recommended it for similar projects.

6. RELATED WORK

To handle semantic heterogeneities in data integration, an ontology-driven approach has been suggested [3, 13]. 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 the schemas of the constituent databases and to serve as the format in which queries are posed by the user.

IGO (Interoperable Geospatial Object) [7], 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 that contains methods to access the local data and an agreement file to hold the relevant metadata.

ISIS (Interoperable Spatial Information System) [12] is a semantic mediation approach to GIS interoperability and proposes a multi-agent architecture and a multi-level data

[1] http://www.elzaresearch.com/landv/axiomap_ie5.html

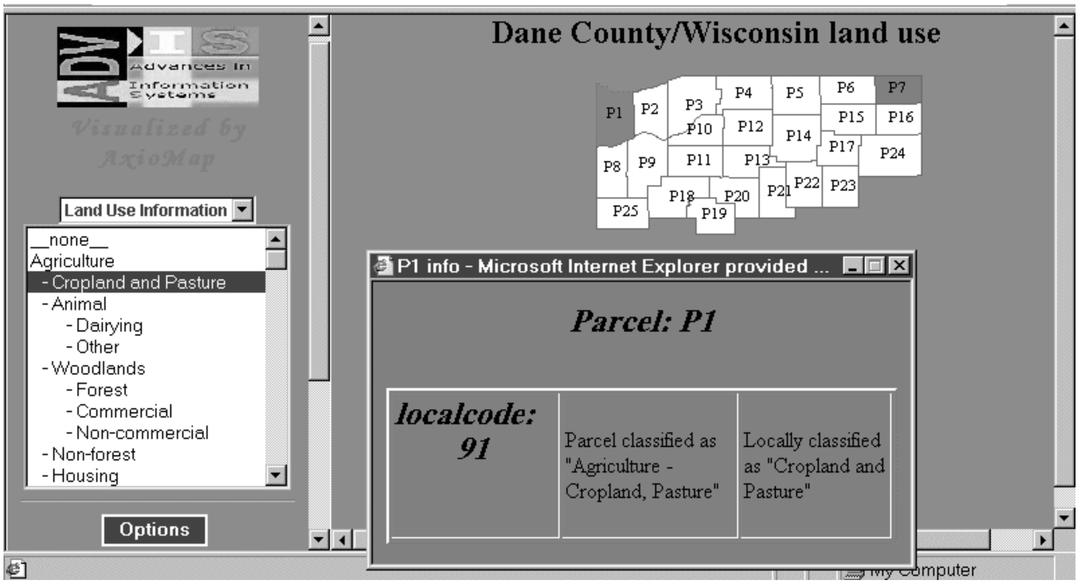

Figure 4: End-user query interface using interactive maps.

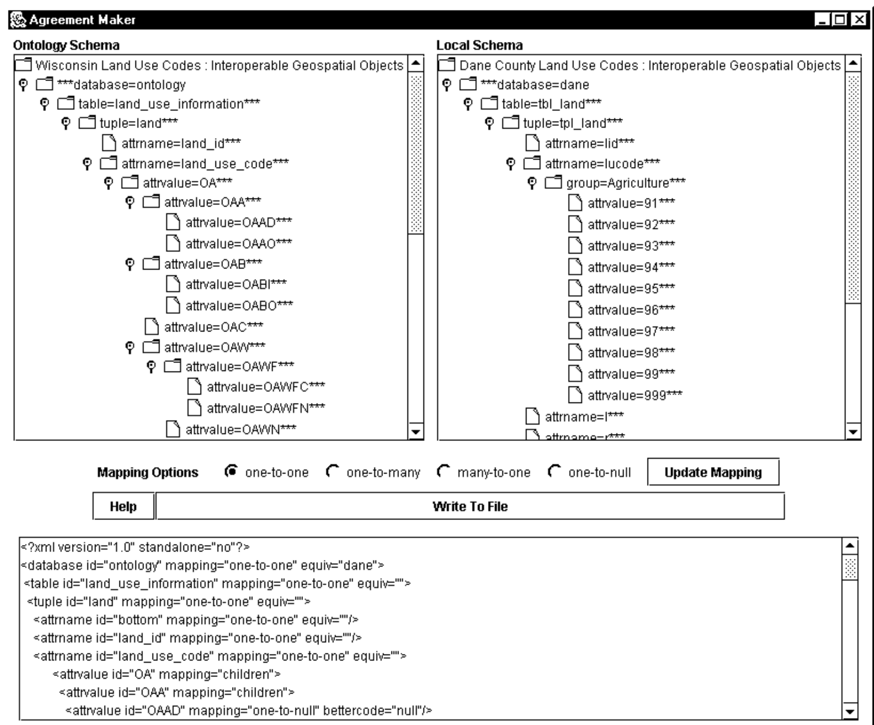

Figure 5: User interface for creating agreements.

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. While we based our architecture on the one of ISIS, the main difference is that in ISIS the methods in the cooperation agent are procedural, while in our approach the data access mechanisms are declarative. This enables new databases to be added to the system with minimal effort.

The approach in [4] used an application to visualize US election results to illustrate the problems in handling different XML representations. A framework of classes was developed to treat XML documents as graphs and collect data by executing XPath expressions against these graphs. A mediated schema was developed and lookup files were created for each state's XML representation to encode the relations to the mediated schema. Due to the fixed 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 an arbitrary number of hierarchical levels.

7. CONCLUSIONS

Due to the increase in complexity and differences in representation of geographic data sources, the process of integrating such data has significant 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 the relations between the entities in the ontology and those in 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 to be integrated into the Wisconsin Land Information System (WLIS).

The practical impact of our approach will be immediately felt in the WLIS prototype query system, which currently can only support very general queries, such as "List all agricultural parcels in Dane County". Using our approach, as more levels of land use codes are represented in the ontology, more specific queries can be supported.

One-to-one mappings between tables, tuples and attributes were handled. If more complex mappings are identified, new query execution mechanisms need to be devised. Our mappings between attribute values assume two hierarchies (the ontology and the local schema) where no inversion of the order occurs. Such mappings can therefore be represented by XPath expressions. While we have not encountered other kinds of mappings so far, that possibility might occur in the future. Therefore, we are extending our framework to handle more expressive queries than those that can be represented by XPath expressions. A natural candidate language is XSLT [5] because of its expressiveness and the availability of fully implemented APIs (e.g., within Apache Xalan). Furthermore, inference mechanisms can be added to the representation of the ontology, which may be helpful in rewriting queries and optimizing their execution [2].

8. ACKNOWLEDGEMENTS

We would like to thank Chaitan Baru and Ilya Zaslavsky for allowing us to use the Axiomap system and Sathish Kumar Murugesan for his help with the implementation.

This research was supported in part by the National Science Foundation under Award EIA-0091489 and by the Advanced Research and Development Activity (ARDA) and the National Imagery and Mapping Agency (NIMA) under Award 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 NIMA or the U.S. Government.

9. 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. Overcoming the Semantic and Other Barriers to GIS Interoperability. *International Journal of Geographical Information Science*, 12:299–314, 1998.

[4] P. W. Calnan and I. F. Cruz. Object Interoperability for Geospatial Applications. In *Proceedings of the First Semantic Web Working Symposium*, pages 229–243, 2001. http://www.semanticweb.org/SWWS/.

[5] J. Clark. XSL Transformations (XSLT) Version 1.0. W3C Recommendation, 1999. http://www.w3.org/TR/xslt.

[6] J. Clark and S. DeRose. XML Path Language (XPath) Version 1.0. W3C Recommendation, 1999. http://www.w3.org/TR/xpath.

[7] J. Doughty, P. Jones, J. Nolan, and S. Hirsch. Interoperable Geospatial Objects. In *GeoComputation '99*, 1999.

[8] F. Fonseca and M. Egenhofer. Ontology-driven Geographic Information Systems. In *7th ACM Symposium on Advances in Geographic Information Systems*, pages 14–19, 1999.

[9] M. Gahegan. Characterizing the Semantic Content of Geographic Data, Models and Systems. *Interoperating Geographic Information Systems*, pages 71–84, 1999.

[10] N. Guarino. *Formal Ontology in Information Systems*. IOS Press, 1998.

[11] V. Kashyap and A. Sheth. Semantic Heterogeneity in Global Information System: The Role of Metadata, Context and Ontologies. In M. Papazaglou and G. Schlageter, editors, *Cooperative Information Systems: Current Trends and Directions*, pages 139–178. Academic Press, 1998.

[12] E. Leclercq, D. Benslimane, and K. Yetongnon. ISIS: A Semantic Mediation Model and an Agent Based Architecture for GIS Interoperability. In *International Database Engineering and Applications Symposium (IDEAS)*, pages 87–91, 1999.

[13] 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 Publishers, 1998.

[14] 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*, pages 115–121, 2002.