XML Database Integration for Visualizing US Election Results

Isabel F. Cruz*, Afsheen Rajendran, and William Sunna

Department of Computer Science
University of Illinois at Chicago
851 South Morgan Street (M/C 152)
Chicago, Illinois 60607-7053

Abstract

In order for data from XML heterogeneous data sets to be viewed and analyzed using a common application, we propose the mapping between each schema and a central schema. The demo uses the example of an application to visualize the US Presidential election results, to illustrate our approach. To minimize the need for human intervention and code maintenance, a declarative approach is used. XML documents are treated as graphs and the queries posed by the user are converted to XPath expressions. The results of the XPath expression are collected and displayed to the user, using interactive maps in a web browser.

1 Introduction

This demo implements an approach to integration of data in heterogeneous XML databases, developed as a part of a framework for geospatial data visualization applications. Applications using this framework should be able to seamlessly pull data from multiple XML data sources. To serve as a unifying ground for the integration of disparate schemas, the concept of a *geospatial authority* is used—authoritative geospatial information about the region being covered by a given application is collected in an XML document, which serves as the source for geospatial relationships at the core of the application.

Each state stores its election results in a different format leading to data heterogeneity. To analyze the election results at the national level, data needs to be retrieved from all states and presented to the user in a common format.

To achieve flexible and dynamic integration of data, minimal human intervention and code restructuring is required. Hence, a declarative approach is used for integrating data from multiple XML sources.

In our demo, the data is presented to the user in interactive maps in a web browser, using the Axiomap [Elz01] visualization extension to the ArcView GIS software. The users can search for a geographic area in the United States, choose the data to display, and then view it on a map of the region. Data can be displayed at multiple granularities, depending on what the user wants to see (e.g., the user can display a map of Wisconsin with county boundaries showing election results for each county and then zoom in on a map of Ashland County showing election results for each municipality).

The framework of classes used for integrating the data is implemented in Java 1.3 and uses Apache's Xalan-Java 2 API for XML and XPath [CD99].

2 Integration Framework

For the purposes of query refinement and schema mapping, it is useful to deal with XML documents as a graph of elements. Our API can retrieve an element's children, parent, siblings, etc., and perform different kinds of aggregation [CC01]. XPath provides a simple way of expressing a path through a document tree to select a set of nodes. When a path expression is evaluated, a set of nodes relative to a context node is selected.

The API for our integration framework consists of a number of core classes that allow applications to treat XML databases as graphs and to evaluate XPath expressions against a document, to perform inter-document lookups and collect the relevant nodes from the XML graphs. Classes are also provided to treat the nodes as data of the appropriate type, to enable aggregation in queries.

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

2.1 Example

In the election results scenario, the geospatial authority is an XML document, which contains geographic information about the area being covered in the United States. The DTD for the geospatial authority (the central schema) is shown below:

```
<!ELEMENT state (county+)>
<!ELEMENT county (municipality+)>
<!ELEMENT municipality EMPTY>
<!ATTLIST state name CDATA #REQUIRED>
<!ATTLIST county name CDATA #REQUIRED>
<!ATTLIST municipality name CDATA #REQUIRED>
```

The classes in the framework are used to retrieve data according to the geospatial authority. There are also documents for each state's election results. Potentially each state can have a different DTD. Next we show the DTD (the local schema) for the election results from Wisconsin.

WI_results.dtd

```
<!ELEMENT state         (county+)>
<!ELEMENT county        (municipality+)>
<!ELEMENT municipality  (wardgroup+)>
<!ELEMENT wardgroup     (candidate+)>
<!ELEMENT candidate     EMPTY>

<!ATTLIST state         name    CDATA #REQUIRED>
<!ATTLIST county        name    CDATA #REQUIRED>
<!ATTLIST municipality  name    CDATA #REQUIRED>
<!ATTLIST wardgroup     name    CDATA #REQUIRED>
<!ATTLIST candidate     name    CDATA #REQUIRED
                        votes   CDATA #REQUIRED>
```

The mapping between the central and local schemas is achieved by defining a lookup mechanism, which consists of the predicates, arguments, and the XPath expression to be used for collecting data from the XML file containing the Wisconsin results.

2.2 Schema transformations

The entity "municipality" in the geospatial authority maps to the entities "municipality" and "wardgroup" in the local schema. This difference is handled by the XPath expression, which encodes the structural relationship between "municipality" and "wardgroup".

Simple queries can be handled using XPath expressions. However, when the queries are complex, they may need arbitrary transformations between the local and central schemas, a more expressive query language is needed. This language can handle graph queries beyond the path queries that can be expressed by XPath. Changing the current application to handle graph queries will simply involve changing the lookup file associated with a local dataset. Our graph queries are a subset of the queries in the G^+ query language [CMW87] and can express aggregations along paths in a graph [CN89].

2.3 Semantic transformations

In this application, as the queries were pre-defined, it was possible to statically include XPath expressions in the lookups for the entities state, county and municipality. In other advanced applications handling geospatial data, it is not possible to define queries (or the lookups) statically. To create dynamic lookups, semantic information in addition to the structural information provided by XML files is needed.

Therefore, we encode the relationships between the local and central schemas in terms of *agreements*. An agreement is created for each local schema, which has to be accessed using the logical view provided by the central schema. The agreements will encode the underlying semantic information. They will help in creating dynamic mappings between varying number of entities in both schemas. The mappings should be able to handle simple, spelling conventions (e.g., Foxboro vs. Foxborough) to complex structural differences. They can also help in inferring new relations based on the existing ones to help in query optimization [BCV99].

If information from a large number of local schemas are to be integrated (e.g., information from all counties in a state), data from lower levels of the schema from a county, will be aggregated into a single entity in the central schema. On the other hand, if data from two counties are to be compared, the resolution

of data will be higher. As in these two cases, the central schema will be different, and therefore we need to use different agreements. In the future, we will be investigating the automatic or semi-automatic generation of the agreements [BM01] to facilitate dynamic querying options, especially when the schemas involved become much larger than the ones we are currently considering.

3 User Interface

The user interface component displays the results using interactive maps in a browser. Initially the user is shown a page with the US map with selectable state areas, as shown in Figure 1. When a state is selected, the browser will direct the user to an analysis of its election results.

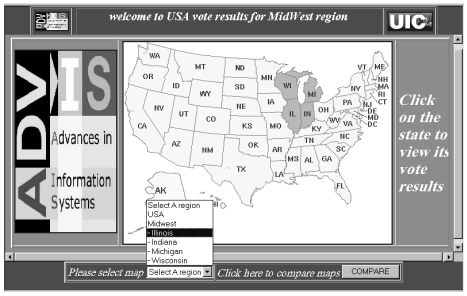

Figure 1: The US Map shown in the opening screen.

For the user convenience, a tool bar is added at the bottom that allows the user to choose which state map to go to. The user should be able to skip between states without going to the original US map. Using the Axiomap display and visualization features, the user can view the vote results for a chosen candidate or the total votes in each county in that state.

The Axiomap display tool allows for two variables to be engaged in an arithmetic calculation. In the map of the state of Wisconsin shown in Figure 2, the number of votes of candidate Gore is displayed relatively to the total votes. In the county of Adams, for example, 52% of the votes went to candidate Gore.

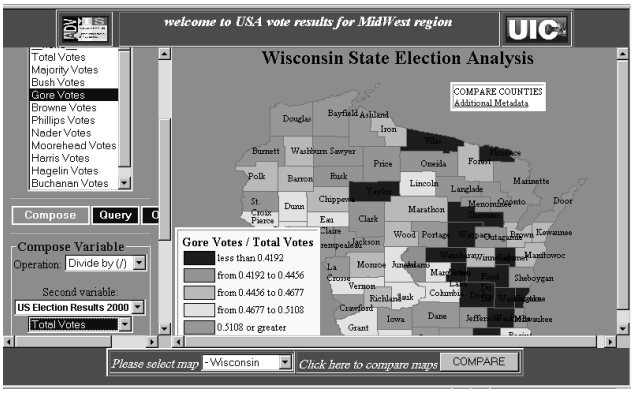

Figure 2: Election results for the state of Wisconsin.

In the state map shown in Figure 2, the user can select a county to view its detailed election analysis. For example, when the county of Adams is selected for viewing, both the map and the elections results are displayed as shown in Figure 3.

We also allow for the comparison of election results in two states, as shown in Figure 4.

Acknowledgments

We would like to thank Chaitan Baru and Ilya Zaslavsky for the Axiomap system. We would also like to thank Nancy Wiegand and Steve Ventura for discussions. This research was supported in part by the National Science Foundation under award EIA-0091489.

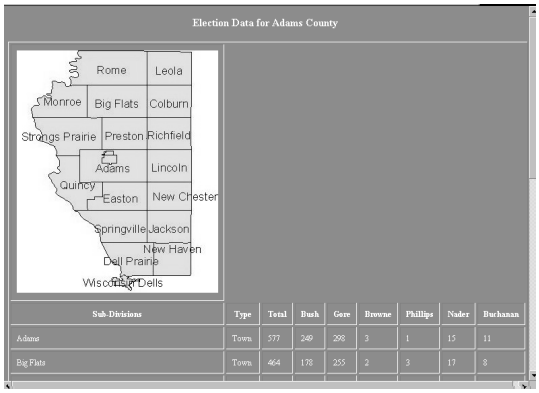

Figure 3: Election results for the county of Adams.

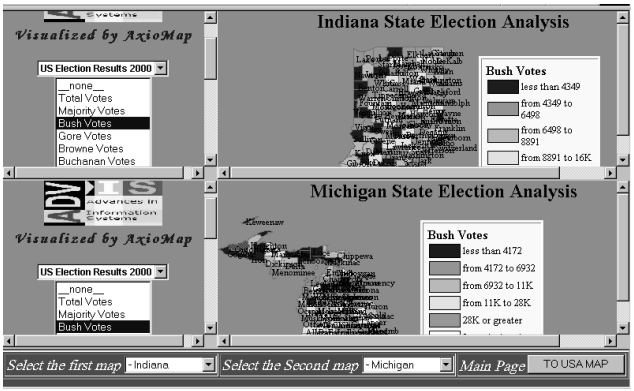

Figure 4: Comparison of the election results between the states of Indiana and Michigan.

References

[BCV99] Sonia Bergamaschi, Silvana Castano, and Maurizio Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, 1999.

[BM01] Jacob Berlin and Amihai Motro. Database Schema Matching Using Machine Learning with Feature Extraction. Technical Report ISE-TR-01-06, George Mason University, 2001.

[CC01] Paul Calnan and Isabel F. Cruz. Object Interoperability for Geospatial Applications. In *Proceedings of the First Semantic Web Working Symposium*, pages 229–243, Stanford, CA, 2001.

[CD99] James Clark and Steve DeRose. XML Path Language (XPath) Version 1.0. W3C Recommendation, 1999.

[CMW87] Isabel F. Cruz, Alberto O. Mendelzon, and Peter T. Wood. A Graphical Query Language Supporting Recursion. In *ACM SIGMOD*, pages 323–330, 1987.

[CN89] I. F. Cruz and T. S. Norvell. Aggregative Closure: An Extension of Transitive Closure. In *IEEE International Conference on Data Engineering*, pages 384–390, 1989.

[Elz01] Elzaresearch. Axiomap: Application of XML for Interactive Online Mapping, 2001.