User Interfaces for Geo-Spatial Applications

William G Sunna
ADVIS research group
wsunna1@uic.edu

CS 597 (MS Project research)

Advisor: Professor Isabel F. Cruz
Table of contents

- Part I (user interface for viewing USA election results) 2
  - Introduction 3
  - System overview 4
  - User interface 6
- Part II (the agreement maker interface for mapping heterogeneous data) 9
  - Introduction 10
  - Overall system description 12
    1. Query interface 12
    2. Query processor 12
    3. The agreement maker 13
  - Using the agreement maker 15
  - Choosing mapping types 18
  - Mapping feedback and operations 20
  - Improvements over the first version 24
- Future work 25
- Acknowledgments 26
- Appendix A (U.S. Elections user’s study) 27
- Appendix B (The agreement Maker User’s Study) 29
- Appendix C (Digital Government Conference 2002, California) 33
- Appendix D 34
- References 35
PART I

User interface for viewing USA election result on the web
Introduction

Visualization is known to be much easier and practical to the humans than reading essays and articles. People usually prefer visualizing data rather than analyzing figures and numbers. In this project, a software tool was used to implement a visual interface for viewing USA presidential election results on the web. Axiomap is a piece of software that was developed by ELZA RESEARCH to display maps that were developed using Arcview software on the web. Axiomap provides many useful features that will be discussed later in this report.

One of the great challenges to computer scientists is the heterogeneity of database systems. In case of presidential election results, various states use different ways to model the election data. In this project, a system is built to integrate the various heterogeneous databases and present them to the user in a uniform format.
System overview

Results
Browser

JAVA API
Using declarative lookup mechanism

XML
DATABASE

USER
The Overall system [1] is composed of the following subsystems:

A. **User Interface**: The user interface is an interactive web page containing navigation tools to view several maps and retrieve election results. The user interface was built using Javascript and Axiomap software.

B. **JAVA API**: is a query processing system that uses declarative lookup mechanism [1]. A detailed description of this system is described in the first reference attached to this report.

C. **XML Databases**: is a pool of heterogeneous databases that include information on the presidential results. Each database has its own schema and level of granularity; for example, the state of Wisconsin has these levels of authorities: state, county, ward (municipality). On the other hand, the state of Illinois has these levels of granularity: state and county. The voting results are stored according to the geospatial authority of each state [1].
User Interface

The user interface component displays the results using interactive maps in a browser. Initially the user is shown a page with the USA 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.

![USA Map](image)

**Figure 1:** the USA 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 map of a state or region to go to. The user should be able to skip between states without going to the original USA 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.
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.

Figure 2: election results for the state of Wisconsin.
Figure 3: Adams county election results.

Figure 4: comparison between two states.
PART II

The agreement maker interface for mapping heterogeneous data
Introduction

Building a system that integrates multiple heterogeneous databases can be a challenging task. We take the land use system in the state of Wisconsin as an example. Each county (local authority) in the state of Wisconsin is divided into parcels, and each parcel is given a code that describes its land use. In the county of Dane for example, parcels with land use code “91” are classified as cropland pastures. Counties (local authorities) adopt different classification schemes and use different codes to describe their parcels leading to heterogeneity. The table below demonstrates how four different authorities use different codes to refer to semantically similar concept. We see that the four different authorities use different terminology in describing both the attribute and the code. All the attributes in the table refer to “land use” and all the codes refer to the fact that any parcel with such code is used for some kind of agricultural purposes.

<table>
<thead>
<tr>
<th>Authority</th>
<th>Attribute</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dane County</td>
<td>Lu_code</td>
<td>91</td>
<td>Cropland Pastures</td>
</tr>
<tr>
<td>Racine County</td>
<td>Tag</td>
<td>811</td>
<td>Cropland Pastures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>815</td>
<td>Other Agriculture</td>
</tr>
<tr>
<td>Eau Claire county</td>
<td>Lu1</td>
<td>AA</td>
<td>General Agriculture</td>
</tr>
<tr>
<td>City of Madison</td>
<td>Lu_4_4</td>
<td>8110</td>
<td>Farms</td>
</tr>
</tbody>
</table>

Building a central ontology is important to achieve database integration. The central ontology should be able to express the concepts in all local ontologies. The agreement maker is a software tool that helps expert database designers to map the central ontology to a local ontology and generate an agreement. In this report I will discuss the overall system and then concentrate on the agreement maker software.
Figure 5: overall system.
Overall system description

1. Query Interface:

The query interface is a web page that uses the Axiomap software. The web page displays interactive maps that are divided into parcels. The user can find out what parcels correspond to what land use by clicking the land use description on the left panel. The parcels with that land use will be highlighted. The user can also click on any parcel and get a pop up window with information on that parcel’s land use. In the Figure 6 below, the user was looking for parcels corresponding to cropland and pasture and got back that both parcel P1 and parcel P2 are used for cropland and pasture. The user also clicked on parcel P1 and got back a pop up window which displayed the local classification of P1.

![Figure 6: query interface.](image)

2. Query Processor:

The query processor uses declarative agreements (refer to [2] in references section) to process the user’s query. It is connected to multiple heterogeneous databases, each database has a local schema that has been mapped to the central schema using the agreement maker. The agreement maker generates agreement files that are used by the query processor to process the queries.
3. The agreement Maker:

The first version of the agreement maker was developed by Afsheen Rajendran and Satish Kumar using Java language as a part of an ADVIS group project in the University of Illinois at Chicago. The first version did not provide many features and was very basic. I had to rebuild the agreement maker to incorporate many more features that facilitate the task of the user. A comparison will be made between the first version and the latest version if the agreement maker later in this report.

The agreement maker takes two XML files as an input, one represents the local schema and the other represents the central (ontology) schema. The database expert will map the concepts of the central ontology to the local one and generate an agreement file that will be used by the query processing system.

The central schema is put together by a database expert. It should be taken into consideration that all local schemata can be expressed in terms of the central schema as much as possible. The central schema is an XML document like the one shown in Figure 7.

```
<?xml version="1.0" standalone="no" ?>
<database id="ontology">
  <table id="land_use_information">
    <tuple id="land1">
      <attrname id="land1_id" />
      <attrname id="land1_use_code" />
      <attrvalue id="OA" exp="Agriculture">
        <attrvalue id="OAA" exp="Agriculture - Animal" />
        <attrvalue id="OAAD" exp="Agriculture - Animal - DairyIng" />
        <attrvalue id="OAAO" exp="Agriculture - Animal - Other" />
      </attrvalue>
      <attrvalue id="OAB" exp="Agriculture - Building" />
      <attrvalue id="OAB1" exp="Agriculture - Building - Industrial Implements" />
      <attrvalue id="OABO" exp="Agriculture - Building - Other" />
    </attrvalue>
    <attrvalue id="OAO" exp="Agriculture - CropLand and Pasture" />
    <attrvalue id="OAW" exp="Agriculture - Woodlands" />
    <attrvalue id="OAWF" exp="Agriculture - Woodlands - Forest" />
    <attrvalue id="OAWP" exp="Agriculture - Woodlands - Forest - Commercial" />
    <attrvalue id="OAWFN" exp="Agriculture - Woodlands - Forest - Non-commercial" />
    <attrvalue id="OAWN" exp="Agriculture - Woodlands - Non-forest" />
    <attrvalue id="OAH" exp="Agriculture - Housing" />
    <attrvalue id="OAHF" exp="Agriculture - Housing - Farm Residence" />
    <attrvalue id="OAHO" exp="Agriculture - Housing - Other" />
    <attrvalue id="OAV" exp="Agriculture - Vacant Land" />
    <attrvalue id="OAVI" exp="Agriculture - Vacant Land - FarmLand" />
    <attrvalue id="OAVN" exp="Agriculture - Vacant Land - Non-farmLand" />
    <attrvalue id="OAT" exp="Agriculture - Water Supplies" />
    <attrvalue id="OAS" exp="Agriculture - Fishing and related" />
  </tuple>
</table>
</database>
```

Figure 7: part of a central schema XML file

The agreement maker parses the local schema (Figure 8) and the central schema (Figure 7) XML files using Xalan and Xerces Java API’s and displays each one as a tree structure in the user interface (Figure 11). The user then selects nodes from each tree and maps them to each other. After accomplishing the task of mapping, the user generates an XML agreement file containing all the mapping information (Figure 17).
Figure 8: part of a local schema XML file for Dane county
Using the agreement maker

The user will start with an interface like the one shown in Figure 9. The system will ask the user load both the local schema (local file) and the central schema (Ontology).

![Figure 9: initial state of the user interface](image)

The user can load the files by clicking on the File icon on the menu bar. A menu will appear with several choices for loading and saving files (Figure 10). The user can choose to click on “Open Ontology File” to load the central schema and then click on “Open Local File” to load the local schema. The user alternatively can click on “Open Project” to resume a previously saved project.
After loading the files, the system will display both schemata in a tree structure (Figure 11). At this point the user can start selecting nodes from both trees by clicking on them. Several nodes can be selected from one tree by using the “Ctrl” or “Shift” keys. After selecting the nodes, the user selects a mapping type by clicking on one of the mapping type radio buttons (Figure 12) or using the Mapping Options Menu (Figure 13). Then the user presses the “Update Mapping” button (Figure 12). The “Update Mapping” button will cause the program to establish a mapping between the chosen entities.
The mapping information will be displayed in the feedback area (Figure 16 and 17) and the mapped nodes will be proceeded by “---＞MAPPED” string to indicate that they have been mapped.

Figure 12: mapping types (Exact, Superset, Subset, Approximate, and No-Mapping)

Figure 13: mapping options menu
Choosing mapping types

There are several options for mapping:

A. **Exact**: this type of mapping is chosen to map one node from the central schema to strictly one node in the local schema that has exact meaning.

B. **Superset**: this type of mapping is chosen when exactly one node in the central schema corresponds to several nodes in the local schema (Figure 14). In the figure we notice that the central ontology land use code OAO (Agricultural-others) is a superset of local classifications of parcels with land use code 99 Agricultural (woodland (non commercial)) and 999 (Agricultural-under construction).

C. **Subset**: this type of mapping is chosen when several nodes in the central schema correspond to exactly one node in the local schema (Figure 15). In the figure we notice that the central land use codes OAAD (Agricultural-Animal and dairying) and OAAO (Agricultural-Animal-other) are subset of the local land use code 92 (Agricultural -Animal Husbandry).

D. **Approximate**: this type of mapping is chosen to map exactly one node from the central schema to exactly one node in the local schema that has an approximate meaning, for example user ID “U_ID” is approximate to user Social Security Number “SSN”.

E. **No Mapping**: Initially all nodes in the central schema do not correspond to any node in the local schema. The user can choose this type of mapping by selecting nodes from the central schema only.
OAO: Agricultural-others

99: Agricultural: (woodland (non commercial))
999: Agricultural-under construction

**Figure 14: superset mapping example**

OAD: Agricultural-Animal and dairying.
OAAO: Agricultural-Animal-other

92: Agricultural -Animal Husbandry

**Figure 15: subset mapping example**
Mapping Feedback and operations

There are two main feedback views to the user to show the mapping actions thus far. The first one is the tabular mapping history like the one shown in figure 16, and the second one is the agreement file thus far (Figure 17). The user switches between the two feedback views by pressing on the button “View XML File/View Table” (Figure 12). The user can also use the menu “View” (Figure 19) to perform the same task.

<table>
<thead>
<tr>
<th>Mapped from</th>
<th>Mapped to</th>
</tr>
</thead>
<tbody>
<tr>
<td>person_name</td>
<td>name</td>
</tr>
<tr>
<td>date_of_birth</td>
<td>dob</td>
</tr>
<tr>
<td>ownerid</td>
<td>owner_scn</td>
</tr>
<tr>
<td>OMA (Agriculture - Animal - Building)</td>
<td>92 (Animal Husbandry (including buildings))</td>
</tr>
<tr>
<td>OMA (Agriculture - Animal - Other)</td>
<td>92 (Animal Husbandry (including buildings))</td>
</tr>
<tr>
<td>OMA (Agriculture - Others)</td>
<td>99 (Vegetation - semi-commercial forest)</td>
</tr>
<tr>
<td>OMA (Agriculture - Others)</td>
<td>999 (Under Construction)</td>
</tr>
</tbody>
</table>

Figure 16: tabular mapping feedback

```xml
<table id="person_information" mapping="No-Mapping">
  <tuple id="person" mapping="No-Mapping">
    <attname id="person_name" mapping="Exact" equiv="name" />
    <attname id="date_of_birth" mapping="No-Mapping"/>
    <attname id="sex" mapping="No-Mapping"/>
  </tuple>
</table>

Figure 17: the agreement file feedback

The agreement maker provides many convenient features like the ability to view the local file and the central file in XML notations. This is done by pressing on the button “XML Trees/Regular Trees” (Figure 12). Figure 18 shows the same tree in regular view and in XML notation view.
Figure 18: regular view (above), XML notation view (below).
Figure 19: the view menu to switch between tabular and XML file feedback.

The user can at any given time undo or redo an action by choosing the undo or redo options from the edit menu (Figure 20). Almost all options come with short cut keyboard keys, For example you can use “Ctrl-z” to perform an undo, and “Ctrl-y” to perform a redo.

Figure 20: the edit menu that contains undo, redo items.

If we go back to the file menu (Figure 10), we notice that the user can save the agreement file using “Save Agreement file/Save Agreement File As” options that behave like any other standard application. If the user decide to finish the work latter, he/she can use the “Save Project/Save Project As” options. Upon saving the project, a file is created with extension “prg”, this file contains special codes that will be verified by the agreement maker when it attempts to open it later. The project file when opened will restore the environment exactly similar to the time it was created.

The agreement maker provides a help menu (figure 21) that contains an options on how to use the software. It also has an option “About Agreement Maker” that will display the product information.
**Figure 21:** the help menu

**Figure 22:** the agreement maker with agreement XML file feedback
Improvements over the first version

The first version of the agreement maker (Figure 23) did not support many features. The user could not load any files but instead has to put the files in a certain directory and give them a certain name. The first version did not support the tabular feedback and did not support the convenience of undoing/redoing the mapping operation. Saving and loading a project was not available for the user. The first version did not cover some mapping options and used inaccurate terminology for mapping. The latest version proved to be visually more appealing by using several icons and a fancy menu bar, it also provided an additional view of the XML trees. From the implementation point of view, the layout was completely changed to handle re-sizing and switching in views as described above.

Figure 23: the first version of the agreement maker
Future Work

Currently the agreement maker handles hierarchies containing small number of nodes like thirty nodes for example; trying to map larger schemata may be very tedious for the user. Therefore the ADVIS group is currently researching methods to semi automate the mapping process. The challenging problem is that the concepts that are being mapped do not carry proper English words but merely codes and abbreviated words.

Alignment in mapping is another issue that ADVIS group is still researching. The idea of Alignment is to establish a connection between a node in the central schema to one that semantically corresponds to it in the local schema. These two nodes are said to be aligned. The mapping between these two nodes is decided by a deduction algorithm based on the relationship of their children.

In the figure above, the user aligned the nodes “A” and “B” and mapped “C” to “G” using exact mapping and “B” to “F” using subset mapping. The mapping between “A” and “D” should be subset by looking at the mapping relations of their children. The deduction algorithm is supposed to detect the mapping of all aligned nodes. There are many other situations ranging from simple like the one above to complicated ones. At this moment, the identification of complicated situations and a deduction algorithm is under research.
Acknowledgments

I would like to thank Professor Isabel Cruz for providing excellent guidance throughout the implementation of the project, and Professor Tadao Murata for participating in the evaluation committee of this project. I also would like to thank the following students: Afsheen Rajendran, Sathish Komar, Huiyong Xiao, Anjli Chaudhry and Sukadev Dandamudi. I also would like to thank all the ADVIS research group at the University of Illinois at Chicago and our colleagues at the University of Wisconsin at Madison: Nancy Wiegand, Steve Ventura, Naijun Zhu, and Dan Peterson. I also would specially thank Ilya Zaslavsky and Chaitan Baru from San Diego Super Computer Center in California for providing the Axiomap Software.
Appendix A:

A user study was conducted to evaluate the Interface for the United State presidential election results. Seventeen graduate students participated and they provided valuable feedback. The results reflect some satisfaction of map visibility and ease of navigation through the various maps. The participants thought that querying the map from the left frame menu was easy and intuitive. For state comparison feature, 89% thought that it is useful while the rest thought it is not. 94% of the participants thought that using the left frame of the interface was trivial, only 6% we confused when using it. Looking at the terminology used, 24% thought it was very precise and 76% rated it as precise and non thought it was ambiguous. As for the use of the colors, 53% thought it was very good and 47% rated it as good. The arrangement of information was thought to be very logical by 18% of the participants and logical by 82%.

The participants complained that it was slow trying to load the interface using slow dialup network service. Another problem was the font visibility when the screen becomes congested with maps specially when comparing two maps in one webpage (Figure 4). Overall, the interface was rated 7.0 on a scale of 10.0, and 94% of the participants would recommend it for similar projects. Table A shows the ratings of some key features of the interface.
<table>
<thead>
<tr>
<th>Table A: the USA election results interface user study results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Map Visibility</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Terminology</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Using Left Frame</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Comparison of States</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Use Of Colors</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Arrangement of Information</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Recommendation</strong></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Appendix B:

A user study was conducted to evaluate the Interface for the agreement maker during its development stages prior to completion. 31 students participated and they provided valuable feedback, which contributed significantly to improving the quality of the product at the end. Figure 24 shows the interface at that stage. Another user study was conducted after the completion of the product with 33 participants showed a slight change of opinions from the first one.

![Figure 24: the agreement maker in developmental stages](image)

In the first study, the participants admired the intuitiveness of the interface, 97% thought it was easy to use the product without many directions. They also liked the tabular feedback mechanism, about 96% thought that it was clear. As for the icon design, 43% thought it was very good and 43% said it was good, only 14% rated it as fair. The arrangement of the displayed information was thought to be logical by 97% of the participants and the terminology used was thought to be precise by also 97%. The undo/redo feature was very clear to 93% of the participants and the switching between tabular feedback and XML file was very clear to 100%. As for recommendation of this software to be used for similar projects, 19% strongly recommend it and 58% recommend it while 19% somewhat recommend it, and only 3% of the participants do not recommend it.

There were many valuable suggestions like making the interface look like a common windows application. Many participants suggested the addition of a menu bar and keyboard shortcuts. Their suggestions were met and 81% of the participants in the second study thought that the use of the menu bar is good. The participants in the first study complained that the window displaying the interface was hard to resize, this was fixed later on. Overall, the software was rated 8.0 on a scale of 10.0.

In the second study, 90% of the participants still thought that the interface is intuitive. As for tabular feedback, 91% thought it was clear. As for the icon design, 30% rated it as very good and 54% rated it as good while only 16% rated it as poor. The arrangement of the displayed information was thought to be logical by 94% of the
participants and the terminology used was thought to be precise by 88%. The undo/redo feature was very clear to 94% of the participants and the switching between tabular feedback and XML file was clear to 82%. As for recommendation of this software to be used for similar projects, 21% strongly recommend it and 55% recommend it while 24% somewhat recommend it.

The participants in the second study, in their written comments, showed admiration in the product in many ways. They liked that they can see what actions they take by the feedback mechanism, and thought the interface to be very visual and easy to learn. Other participants thought the icons used for mapping options were expressive in meaning and helped them decide their choice of mapping. Others stated that the interface was simple and consistent with other applications from a design point of view. Some participants ran into a font clarity problems when running the software in UNIX environment, they said that the fonts did not display well. Other participants criticized the error messages design and demanded more sophisticated error dialogs. Participants suggested many ideas to improve the product, an addition of a tool bar would give the interface even a more professional look. Others suggested a selective undo where the user can choose any mapping action to be undone from the action table instead of sequentially undoing what came after it. Another idea, which may be undesirable to some, is to give the user the power to change the XML agreement file manually from the interface, this idea if implemented assumes that the user is unlikely to make an error in changing the format of the agreement file. Some suggested adding a print method to print the agreement file thus far and others said a drag and drop interface would be a good idea. Overall, the software was rated 7.8 on a scale of 10.0.

Comparing the first user study to the second one, we would like to note that the software became more sophisticated and therefore less intuitive. Many features were added after the first study especially the ability to save and load projects. These new capabilities put a burden on the user to understand on how to operate the software. Many problems were corrected and complaints against them disappeared in the second study, for example the interface window is resizable now. The recommendation for using the product stayed almost the same and the rating of the interface did not change very much. Table B shows the ratings of some key features of the software from the first user study and Table C shows the ratings of the same features from the second user study.
<table>
<thead>
<tr>
<th>Intuitiveness</th>
<th>Very Intuitive</th>
<th>Somewhat Intuitive</th>
<th>Somewhat not intuitive</th>
<th>Not Intuitive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>55%</td>
<td>42%</td>
<td>3%</td>
<td>0%</td>
</tr>
<tr>
<td>Terminology</td>
<td>Very Precise</td>
<td>Precise</td>
<td>Ambiguous</td>
<td>Very Ambiguous</td>
</tr>
<tr>
<td></td>
<td>42%</td>
<td>55%</td>
<td>3%</td>
<td>0%</td>
</tr>
<tr>
<td>Use of tabular feedback</td>
<td>Clear</td>
<td>Somewhat clear</td>
<td>Somewhat unclear</td>
<td>Unclear</td>
</tr>
<tr>
<td></td>
<td>77%</td>
<td>20%</td>
<td>3%</td>
<td>0%</td>
</tr>
<tr>
<td>Icon design</td>
<td>Very good</td>
<td>Good</td>
<td>Fair</td>
<td>Poor</td>
</tr>
<tr>
<td></td>
<td>44%</td>
<td>43%</td>
<td>13%</td>
<td>0%</td>
</tr>
<tr>
<td>Arrangement of displayed Information</td>
<td>Very Logical</td>
<td>Logical</td>
<td>Illogical</td>
<td>Very Illogical</td>
</tr>
<tr>
<td></td>
<td>71%</td>
<td>26%</td>
<td>0%</td>
<td>3%</td>
</tr>
<tr>
<td>Recommendation</td>
<td>Highly Recommend</td>
<td>Recommend</td>
<td>Not Recommend</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19%</td>
<td>58%</td>
<td>22%</td>
<td></td>
</tr>
</tbody>
</table>

*Table B: the agreement maker user study results at developmental stages*
<table>
<thead>
<tr>
<th>Intuitiveness</th>
<th>Very Intuitive</th>
<th>Somewhat Intuitive</th>
<th>Somewhat not intuitive</th>
<th>Not Intuitive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>42%</td>
<td>48%</td>
<td>9%</td>
<td>0%</td>
</tr>
<tr>
<td>Terminology</td>
<td>Very Precise</td>
<td>Precise</td>
<td>Ambiguous</td>
<td>Very Ambiguous</td>
</tr>
<tr>
<td></td>
<td>55%</td>
<td>33%</td>
<td>12%</td>
<td>0%</td>
</tr>
<tr>
<td>Use of tabular feedback</td>
<td>Clear</td>
<td>Somewhat clear</td>
<td>Somewhat unclear</td>
<td>Unclear</td>
</tr>
<tr>
<td></td>
<td>84%</td>
<td>6%</td>
<td>9%</td>
<td>0%</td>
</tr>
<tr>
<td>Icon design</td>
<td>Very good</td>
<td>Good</td>
<td>Fair</td>
<td>Poor</td>
</tr>
<tr>
<td></td>
<td>46%</td>
<td>45%</td>
<td>9%</td>
<td>0%</td>
</tr>
<tr>
<td>Arrangement of displayed Information</td>
<td>Very Logical</td>
<td>Logical</td>
<td>Illogical</td>
<td>Very Illogical</td>
</tr>
<tr>
<td></td>
<td>70%</td>
<td>24%</td>
<td>6%</td>
<td>0%</td>
</tr>
<tr>
<td>Recommendation</td>
<td>Highly Recommend</td>
<td>Recommend</td>
<td>Not Recommend</td>
<td></td>
</tr>
<tr>
<td></td>
<td>21%</td>
<td>55%</td>
<td>24%</td>
<td></td>
</tr>
</tbody>
</table>

*Table C: the agreement maker user study results at final stages*
Appendix C:

I participated on demonstration of the user interface for viewing USA presidential election results in the Digital Government conference. The Conference was held in Redondo Beach, California in May 2002. Many Universities participated in proposing new research ideas and demonstrated various systems. The Digital Government conference is sponsored by National Science Foundation. I demonstrated the system to the program managers of Digital Government: Dr. Lawrence Brandts, and Dr. Valerie Greg. They were impressed, as well as many other attendants, by the system I demonstrated.

Left to right: William, Isabel, [demo screen], Nancy, and Naijun happy after a very successful demo!
Appendix D:

The agreement maker places some mapping restrictions on the user; Error messages will appear if the user violates some mapping rules imposed by the program. Here is a list of the mapping rules:

- The user can not select many nodes in either the central tree or the local tree and map them as exact.
- The user can not select many nodes on the central tree and one or many nodes in the local tree and map them as superset.
- The user can not select one node from the central tree and many nodes from the local tree and map them as subset.
- The user can not select many nodes in either the central tree or the local tree and map them as approximate.
- The user can not map nodes that carry different XML tags. For example a node that is tagged as an attribute name cannot be mapped to a node that is tagged as an attribute value. The user can see a distinction between these nodes by looking at the interface, generally some nodes that are represented by a directory icon like “land_use_code” in figure 11 can’t be mapped with nodes represented by file icon, like “93 (Farm Buildings and Accessories)” in figure 11.
References:

