Available online at www.sciencedirect.com

ScienceDirect

Journal of Visual Languages and Computing
18 (2007) 230–254

Journal of
Visual Languages
& Computing

www.elsevier.com/locate/jvlc

A visual tool for ontology alignment to enable geospatial interoperability[☆]

Isabel F. Cruz*, William Sunna, Nalin Makar, Sujan Bathala

Department of Computer Science, University of Illinois at Chicago, 851 S. Morgan St. (M/C 152), Chicago, IL 60607, USA

Abstract

In distributed geospatial applications with heterogeneous databases, an ontology-driven approach to data integration relies on the alignment of the concepts of a global ontology that describe the domain, with the concepts of the ontologies that describe the data in the distributed databases. Once the alignment between the global ontology and each distributed ontology is established, *agreements* that encode a variety of mappings between concepts are derived. In this way, users can potentially query hundreds of geospatial databases using a single query. Using our approach, querying can be easily extended to new data sources and, therefore, to new regions. In this paper, we describe the AgreementMaker, a tool that displays the ontologies, supports several mapping layers visually, presents automatically generated mappings, and finally produces the agreements.

Keywords: Ontology alignment; Information integration; Geospatial data

1. Introduction

Concepts in geospatial databases are often categorized and described using ontologies. Such ontologies can be created independently by domain experts who have minimum or no communication among them. As a result, similar concepts can be described differently and

[☆] This research was supported in part by the National Science Foundation under Awards ITR IIS-0326284 and IIS-0513553.

*Corresponding author.

E-mail addresses: ifc@cs.uic.edu (I.F. Cruz), wsunna@cs.uic.edu (W. Sunna), nmakar@cs.uic.edu (N. Makar), sbathala@cs.uic.edu (S. Bathala).

1045-926X/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jvlc.2007.02.005

their categorization can result in heterogeneous ontologies. Even in the case where a standard ontology has been established for a particular domain, its customization to particular regions will result in heterogeneous ontologies.

We have proposed an integration framework to facilitate the access to the information that is contained in distributed and heterogeneous databases [1]. Our approach relies on the alignment of ontologies, that is, on establishing mappings among related concepts in two heterogeneous ontologies. When such mappings have been established, we say that the two ontologies are *aligned* or *matched*. We consider two different architectures: a centralized architecture and a peer-to-peer (P2P) architecture. In the former architecture, there is a global ontology. Each distributed ontology is aligned with the global ontology. As a consequence, a query expressed in terms of the concepts of the global ontology can be translated into a query to one of the distributed or *local* databases using the mappings that are established during the alignment process. In the latter architecture, a query to one of the peers, the *source* peer, can be translated into a query to another peer, the *target* peer, provided that the ontologies of the two peers have been aligned. Whichever the architecture, querying can be easily extended to new databases, and therefore to new regions.

In our paper, we present two case studies in the geospatial domain, which illustrate the two architecture types. In the first case, we discuss distributed databases that store land use data in the state of Wisconsin. Land use data categorizes the parcels in a county or municipality according to their usage. For example, a parcel can be used for *agricultural*, *residential*, *industrial*, *recreational*, or one of many other land use purposes. In the state of Wisconsin, each county and each municipality uses different ontologies. For instance, a parcel used for *business services* in the city of Madison is assigned the land use code 324 while a parcel used for the same purpose in the Fitchburg township is assigned the land use code 63. Therefore the query "return the land parcels that are used for business services in the state of Wisconsin" will give rise to many different queries. While this example simply requires a code translation for each different county and municipality, other correspondences can be more complex, for example mapping one code in one ontology to two or more codes in another ontology. In the absence of our integration framework, all those queries would have to be manually generated, whereas by using our framework, only one query needs to be expressed.

In the second case study, we present two wetland ontologies. In this particular case, one of the ontologies was customized from the other ontology so as to better describe a particular region. This case will be used to illustrate a P2P architecture and to make the point that even when standards exist, ontology alignment is needed to automatically propagate queries to databases whose heterogeneity results from customization.

In order to resolve heterogeneities and bridge the gap between distributed systems such as the ones mentioned in our case studies, we propose a multi-layered approach to ontology alignment, whose functionality extends that of our previous work [1–3]. In our approach, mappings are determined semi-automatically, using both automatic and manual methods. To establish such mappings, our approach uses four mapping layers. Three of these layers use automatic methods and the other one uses manual methods. In our previous work there was a single layer, which supported both automatic and manual features, which are a subset of those that are currently supported.

We propose a tool, the AgreementMaker, which implements our multi-layered approach. The AgreementMaker supports a graphical user interface, that displays the two ontologies to be aligned side by side. The mappings among concepts are displayed as straight lines, which can be produced automatically by the tool or can be created or modified manually by the user. In creating the graphical user interface, we took into consideration a variety of issues, including the manipulation of large ontologies, the presentation of the mappings as produced by the different layers, and the customization of the displayed results. Upon aligning the ontologies, our tool generates *agreements* that encode the mappings produced by the four layers and stores them in an *agreement document*. Such mappings will be used by end applications such as those providing querying capabilities across distributed databases. In particular, we show how the obtained mappings are used to display the results of geospatial queries on distributed land use databases.

The rest of this paper is organized as follows. In Section 2, we give an overview of related work in the area. We present the land use and the wetland classification case studies in Section 3. In Section 4, we present the different types of mappings that are implemented by our multi-layered approach to ontology alignment. In Section 5 we describe the architecture of the AgreementMaker and describe issues of user interaction. In addition, we discuss the agreement document and how it can be used to create land use maps. In Section 6, we present the results of aligning several sets of ontologies using our tool in order to validate our multi-layered mapping approach. Finally, in Section 7, we draw conclusions and describe our future work.

2. Related work

The state of the art of ontology and schema alignment methodologies was recently surveyed by Shvaiko and Euzenat [4]. Previously, Rahm and Bernstein surveyed schema matching in databases [5]. In this section, we cover ontology and alignment tools, whose methodologies are close to our own, even if most of them do not focus specifically on the geospatial domain. A notable exception is offered by Fonseca et al. [6]. They introduce an ontology-driven geographic information system (ODGIS), which is used to drive the creation of ontologies that will enable the integration of geospatial data. In our case, we are concerned with establishing mappings among the concepts in existing ontologies, not with designing those ontologies.

Chimaera [7] is a software tool developed by the KSL group at Stanford, which provides tools for merging ontologies and checking the correctness of ontologies. Chimaera is web-based. Its graphical user interface supports a set of commands accessible via spring-loaded menus as well as drag and drop editing. The interface displays the knowledge base being edited and allows for users to check an automated merging procedure by highlighting the classes that require the user's attention.

COMA++ [8] is a schema and ontology mapping tool, which is in many ways similar to our own mapping tool. However, both tools have been developed independently. COMA++ supports an iterative and automatic matching of ontology components and multiple matching algorithms. COMA++ supports multiple ontology and schema formats such as OWL, XSD, and XML. In comparison, we have a similar approach because we also use several matching algorithms in our mapping process. One difference between COMA++ and our tool is that COMA++ can reuse mapping results that were obtained from aligning a pair of ontologies in the alignment of other similar pairs of ontologies. With our tool, such reuse is also possible, but only in deriving new mappings between the same pair of ontologies. Perhaps the main difference between the two approaches is that the development and design of our AgreementMaker tool has been driven by real applications that happen to be in the geospatial domain, whereas COMA++ does not have a similar emphasis.

The MAFRA toolkit is a mapping framework for distributed ontologies which adopts an open architecture in which concept mappings are realized through semantic bridges. A semantic bridge is a module that transforms source ontology instances into target ontology instances. The MAFRA toolkit supports a graphical user interface that provides domain experts with functionalities that are needed for the specification of semantic bridges. Our approach is similar to theirs in the sense that we are using multiple mapping layers, which are comparable to their semantic bridges. However, in the MAFRA toolkit, the ontologies are represented as graphs and in particular cases as trees using the Touch Graph library (http://www.touchgraph.com), whereas our tool represents ontologies in the geospatial domain that are trees. Our trees can therefore be displayed using the outline tree paradigm, which we implemented ourselves so as to have tree manipulation characteristics that are familiar to most computer users.

Falcon-AO [9] is an automatic ontology alignment tool that uses linguistic and graph matching techniques. It is similar to our tool in that it attempts to align ontologies using linguistic similarity between two entities relying on their names, labels, comments and other descriptive information. However, Falcon-AO also relies on a graph matcher, which measures the structural similarity between the graphs that represent the ontologies.

Clio [10] is a graphical tool used for the semi-automatically mapping of relational and XML schemas. In contrast, our mapping tool is mainly intended to match ontologies and therefore supports the mapping of XML and OWL/RDFS ontologies represented in XML, RDFS, OWL, or N3 [11]. Using Clio, the user loads a source schema and a target schema and establishes connections between objects in both schemas graphically. Such connections are referred to as value correspondences, which express how an object or more in the source schema are transformed into a target value. Clio has a mapping engine that incrementally produces database (SQL) queries that realize the mappings implied by the correspondences. The AgreementMaker generates a document that shows the mappings between concepts and can be used in a variety of ways, including in generating database queries.

MapOnto [12], which is inspired by Clio, is a research prototype for mapping between a database schema and an ontology as well as between two different database schemas. MapOnto works in an interactive and semi-automatic manner, taking input from the user for creating simple attribute-to-attribute correspondence and allowing the user to select a set of logical formulas that can be used to establish correspondences between related attributes. These logical formulas are generated by the tool using knowledge embedded in the ontologies. These logical formulas are ordered to suggest to the user the most reasonable mapping between the two models. MapOnto's supports a graphical interface, which is based on Protégé [13]. Unlike our tool, the correspondences between attributes are not represented by lines in the interface, but as logical formulas displayed in a separate pane.

3. Geospatial case studies

In this section, we present two case studies that illustrate the need for ontology alignment so as to enable interoperability among distributed databases. The first case relates to land use codes and the second case relates to wetland classifications.

3.1. Land use code case study

In the first case study, we investigate databases containing the classification of land parcels according to their usage. In this case study, the architecture is centralized. Although our examples are from the state of Wisconsin, similar cases could be found in other states. In Wisconsin, counties and municipalities maintain different land use ontologies for their land parcels. Categories of land use include *Agriculture*, *Commerce*, *Industry*, and *Residence*. Furthermore, each category comprises several subcategories.

Each land use category and subcategory is assigned an alphanumeric or a numeric value called land use code. For example, the land use code for *Business services* in the city of Madison is 324, while it is 63 in the Fitchburg township. This case is particularly interesting because both the city of Madison and the Fitchburg township are in Dane county. Table 1 illustrates the heterogeneity of selected land use codes in the city of Madison and in the Fitchburg township.

While this example simply requires a code translation for each different county and municipality, other correspondences can be more complex. For example, one code in one ontology can correspond to two or more codes in another ontology. Also, attribute names can be different. For example, the attribute name for land use is *Lucode*, *Lu_4_4*, or *Lu1* depending on whether we are considering Dane County, the city of Madison, or Eau Claire County, as illustrated in Table 2. This table also illustrates the issue previously mentioned that one code in one ontology may correspond to more than one code in the other ontology. For example, code 41 in Dane County does not seem to correspond to a single code value in Madison. Likewise, it is possible that a code value in one city may have no correspondence in another city or municipality.

3.2. Wetland ontologies

The second case study relates to wetland classifications and illustrates the use of a P2P architecture. Organizations monitoring wetlands need to share associated data and information. However, the lack of a standard classification has long been identified as an obstacle to the development, implementation, and monitoring of wetland conservation strategies at national, provisional and local levels [14].

Table 1
Heterogeneity of land use codes in Madison and Fitchburg

Land use category	Land use code (Madison)	Land use code (Fitchburg)
One unit residence	1110	111
Multi-unit residence	113	115
Personal services	323	62
General repair services	325	531
Apparel and accessories	315	56
Finance, insurance, and real estate	322	61
Sewage	4841	487
Automobile parking	370	46
Cemeteries	430	76
Vacant lands	9	98

Table 2
Heterogeneity of attribute names and values

Planning authority	Attribute	Code	Description
Dane County	Lucode	41	Railroad, transit.
City of Madison	Lu_4_4	4112	Railroad switching and marshaling yards.
		4113	Railroad terminal (passenger).
		4114	Railroad terminal (freight).
		4115	Railroad terminal (passenger and freight).
		4116	Railroad equipment and maintenance.
		4119	Other railroad transportation, NEC.
		412	Rapid rail transit and street railway transportation.
Eau Claire County	Lu1	PWR	Railroad.

In defining wetlands, the United States adopts the Cowardin system [14] shown in Fig. 1. In contrast, European nations and Canada use the International Ramsar Convention classification (http://www.ramsar.org). Most classifications recognize the need for regionalization because of the variations in climate, geology, soils, and vegetation. Regionalization is designed to facilitate three activities: (1) planning, where it is necessary to study management problems and potential solutions on a regional basis; (2) organization and retrieval of data gathered in a resource inventory; and (3) interpretation of inventory data, including differences in indicator plants and animals among the regions. It can thus be concluded that it is extremely difficult to have a single standard that is adopted by all nations or by geographically spread regions [14].

In this case study, we concentrate on the Cowardin ontology shown in Fig. 1 and on the South African wetland ontology [15] shown in Figs. 2 and 3. Our main goal is to align both ontologies using our multi-layered alignment approach to allow for data and information sharing between the two countries. The Cowardin ontology has five subcategories: *Marine*, *Estuarine*, *Riverine*, *Lacustrine*, and *Palustrine*. The Marine and Estuarine subcategories each have two subcategories called *Subtidal* and *Intertidal*. The *Riverine* subcategory has four subcategories called *Tidal*, *Lower Perennial*, *Upper Perennial*, and *Intermittent*. The *Lacustrine* subcategory has two subcategories, called *Littoral* and *Limentic*. Finally, the *Palustrine* subcategory has no subcategories. The concepts that are represented as leaves of the trees, include *Rock Bottom*, *Unconsolidated Bottom*, *Rocky Shore*, and *Unconsolidated Shore*.

One of the main challenges in aligning automatically the two wetland ontologies shown in Fig. 1 and 2 is the possibility of producing misleading mappings between concepts with the same name, which are, however, classified under non-corresponding categories. For example, the concept *Reef* in Fig. 1 that is classified under *Intertidal* should not be mapped to the concept *Reef* in Fig. 2 that is classified under *Subtidal*. Many other such misalignments could occur in these two ontologies. Therefore, in our ontology alignment approach, we took such situations into consideration, which we will discuss in Section 4.1.

In this case study, both the Cowardin and South African wetland ontologies are represented as XML trees that store the various categories, subcategories, and concepts of the wetland ontologies as XML nodes. We refer to all those nodes as concepts throughout this paper. In our examples, we map concepts from the Cowardin classification to the

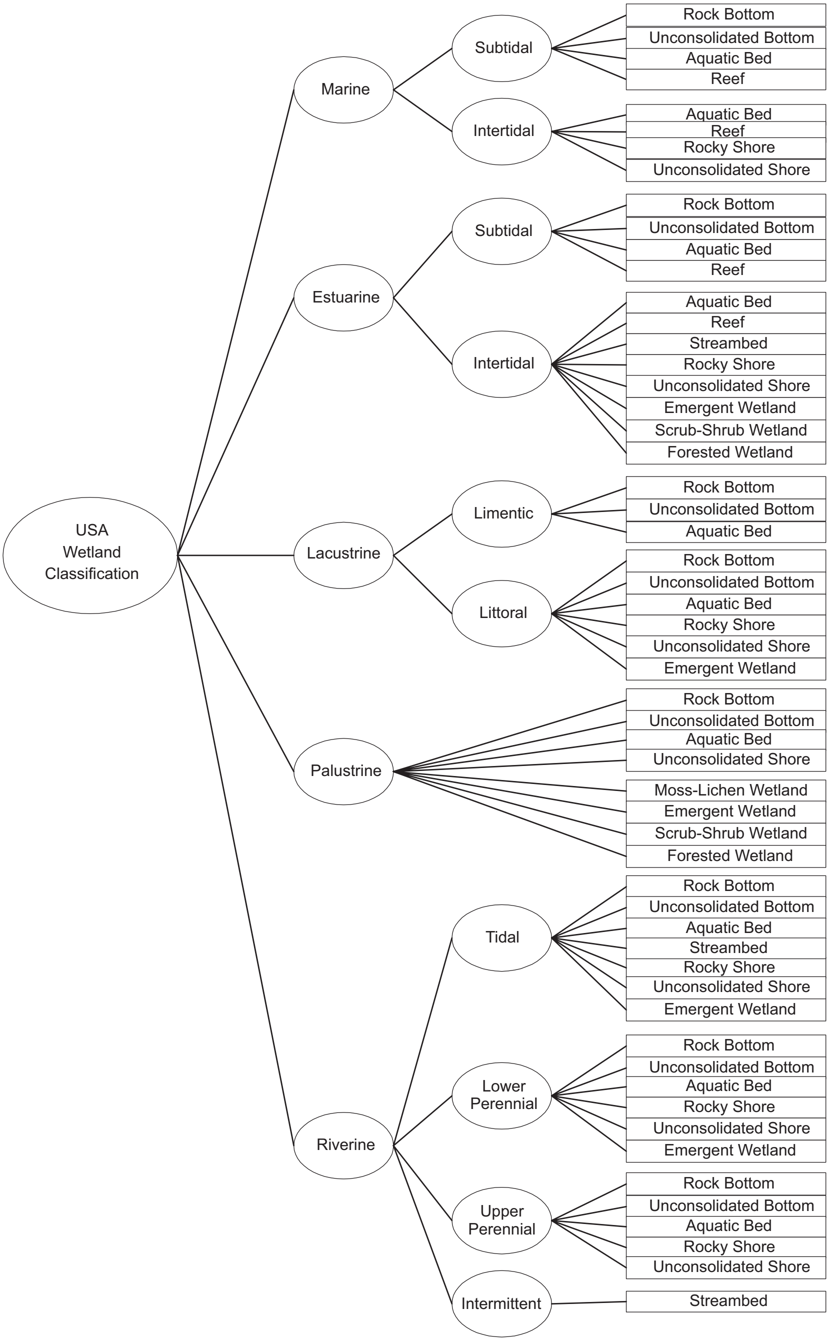


Fig. 1. Cowardin wetland ontology (used in the USA).

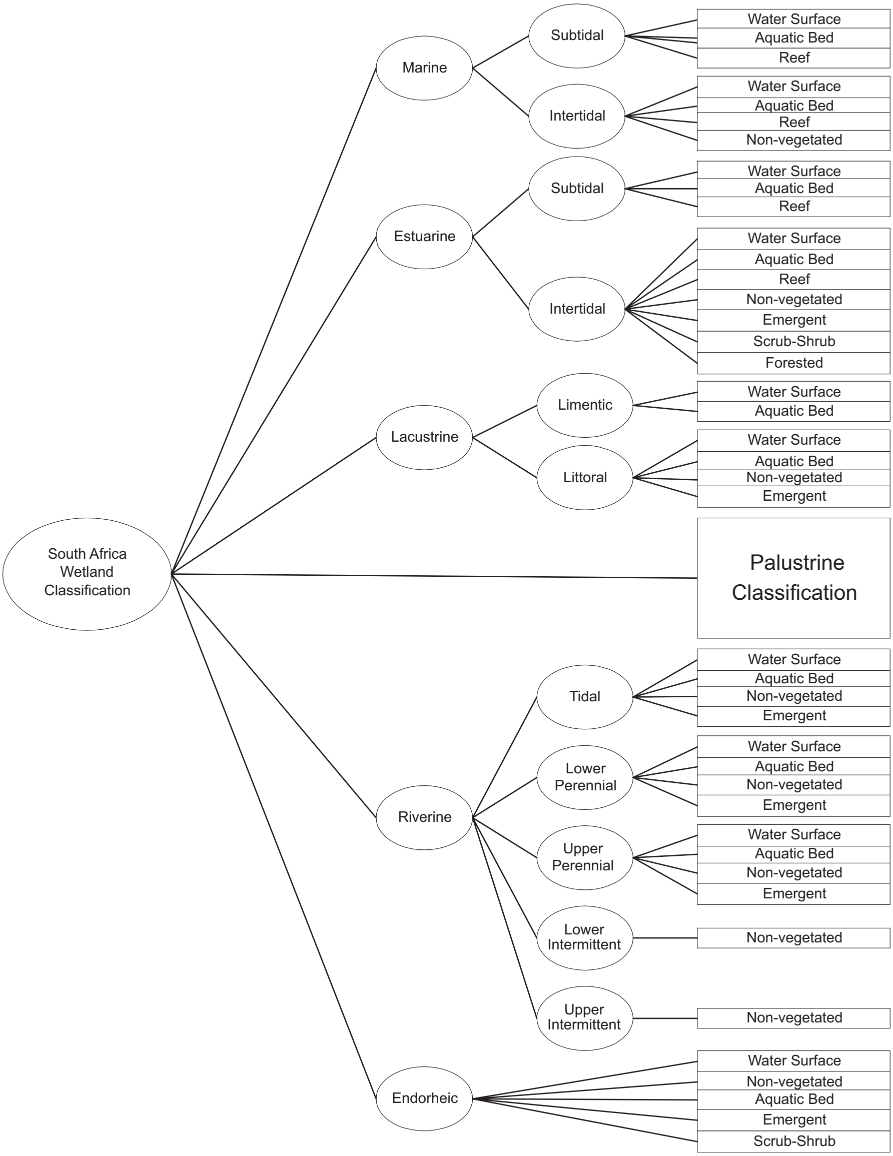


Fig. 2. South African wetland ontology.

South African classification, therefore we refer to the Cowardin classification as the source ontology and the South African classification as the target ontology.

4. Multi-layered alignment approach

In order to align ontologies, we need to identify the semantic correspondences among their concepts. The identification process is achieved by mapping a concept or concepts in one of the ontologies to a concept of concepts in the other ontology based on one or

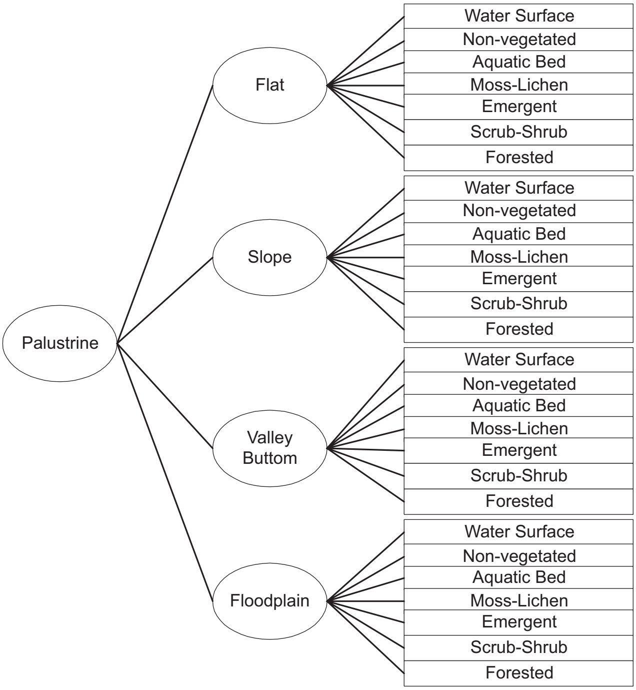


Fig. 3. South African wetland Palustrine ontology.

several matching criteria. For this purpose, we identified four layers of mappings. In each layer, concepts may be mapped differently depending on the matching criterion incorporated in that particular layer. We note that we use several layers so as to enhance the accuracy of the alignment by considering different matching strategies. Also, in this way, the user can be selective in what alignment layers to use and what alignment layers to ignore, depending on the actual ontologies to align.

In the figures associated with our examples, the ontology on the left represents either the global or the source ontology and the ontology on the right represents either the local or the target ontology, depending on the architecture used.

4.1. *Automatic mapping by definition*

In this mapping layer, the user invokes an automatic procedure that compares each concept in the global (source) ontology to each concept in the local (target) ontology according to their definition, as provided by a dictionary. A similarity measure from 0% (no match) to 100% (exact match) between the concepts being compared is returned. If the user does not want to consult the dictionary, the procedure will be performed by comparing only the concept names and any associated descriptions or properties of the concepts.

In this layer, we take into consideration situations where a match between two concepts with the same name can be incorrect as discussed in Section 3.2. In order to eliminate such situations, we take into consideration the path from the concept to the root in determining the similarity. Therefore, the concept *Reef* that belongs to the *Intertidal* wetland subcategory in the source ontology will only match exactly with the concept *Reef* which belongs to the *Intertidal* wetland subcategory in the target ontology as shown in Fig. 4. We note that, in general, the concepts along the two paths may not provide an exact matching. In such cases, the similarity must be computed using a function that returns the similarity between the two paths as a function of the similarities of pairs of concepts along the two paths, so that the most similar pair of paths can be determined.

4.2. Manual mapping by the domain expert

In this layer of mapping, a domain expert maps concepts manually according to the expert's knowledge of the domain. Relations between mapped concepts can take several forms. We refer to such relation forms as mapping types.

Exact a concept in the global (source) ontology is equivalent to a concept in the local (target) ontology.

Subset a relation between a set of concepts in the global (source) ontology and a single concept in the local (target) ontology, where each concept in the global (source) ontology is a subset of the concept in the local (target) ontology.

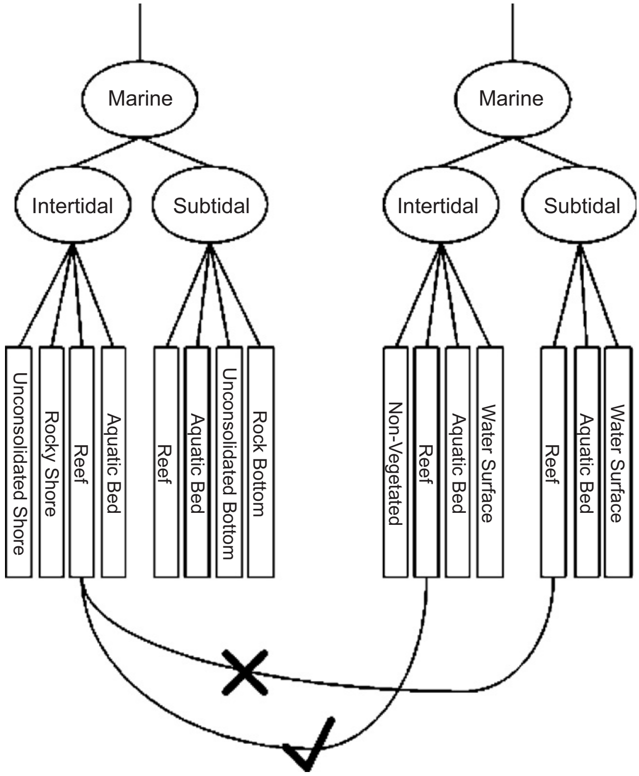


Fig. 4. A case where incorrect mappings may occur if the complete paths are not considered.

Subset complete a subset mapping, where the set of concepts in the global (source) ontology is equivalent to the single concept in the local (target) ontology.

Superset a relation between a single concept in the global (source) ontology and a set of concepts in the local (target) ontology, where the concept in the global (source) ontology is a superset of each concept in the local (target) ontology.

Superset complete a superset mapping, where the single concept in the global (source) ontology is equivalent to the set of concepts in the local (target) ontology.

Comparative a concept or a set of concepts in the global (source) ontology is mapped to a concept or a set of concepts in the local (target) ontology. While such mappings do not fall in any of the above categories because the concepts are more loosely related, users can use this mapping type if the creation of mappings in these circumstances is deemed useful.

4.3. Automatic mapping by context

In this mapping layer, the user can run a procedure that automatically deduces more mappings by considering previously established mappings. Having mapped concepts at the lower leaf level of the ontological tree, this deduction process can potentially map the upper level concepts [3]. The introduction of this layer intends to simplify the task of mapping large ontologies where automatic mappings propagate up the ontological tree.

Fig. 5 shows an example of our deduction process. The user maps the concept *Repair services* in the global ontology to the concept *Recovery services* in the local ontology as an *exact* match. The user then maps the global ontology concepts *Insurance and Finance* to the local ontology concept *Financial*, *insurance*, *and real estate* as a *subset*. The concept *Constructions* in the local ontology remains unmapped in this example. Having performed these mappings, the user may invoke the deduction procedure implemented in this layer. The procedure will consider the mappings among the children of the concept *Commercial*

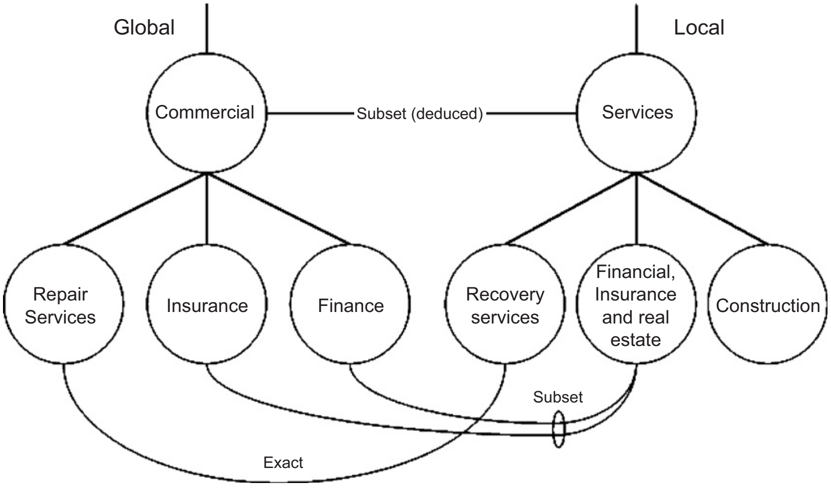


Fig. 5. Context mapping example.

and the children of the concept *Services* to determine if it can deduce the mapping between them. In our example, the deduction procedure will automatically deduce that the mapping between *Commercial* and *Services* is of type *subset*. We note that certain deductions cannot be performed automatically and need therefore to be manually specified [3].

For the deduction procedure to work, we need to establish two assumptions to guarantee its correctness. The first one is that the specialization of a concept in the ontology must be total, that is, that the union of all the concepts under a concept equals the parent concept. The second one is that ‘‘bowties’’ [16], which are inversions in the order of the concepts in a path of one of the ontologies (as compared to the order of the concepts in the corresponding path of the other ontology), do not occur.

4.4. *Automatic mapping by consolidation*

After performing automatic and manual mappings in the previous layers, some concepts may be mapped only in one layer while other concepts may be mapped in more than one layer. The mapping by consolidation layer was introduced to summarize the results of mappings from the previous layers and to resolve any conflicts between the layers that may map the same concept differently. In this layer the user determines the importance of the previous layers by assigning a priority to each one. In this way, if there is a conflict in the case where a concept was mapped by more than one layer, the mapping information from the highest priority layer is the one to be considered. All the mappings are kept, therefore priorities can be changed without recomputing the mappings.

5. Agreement maker

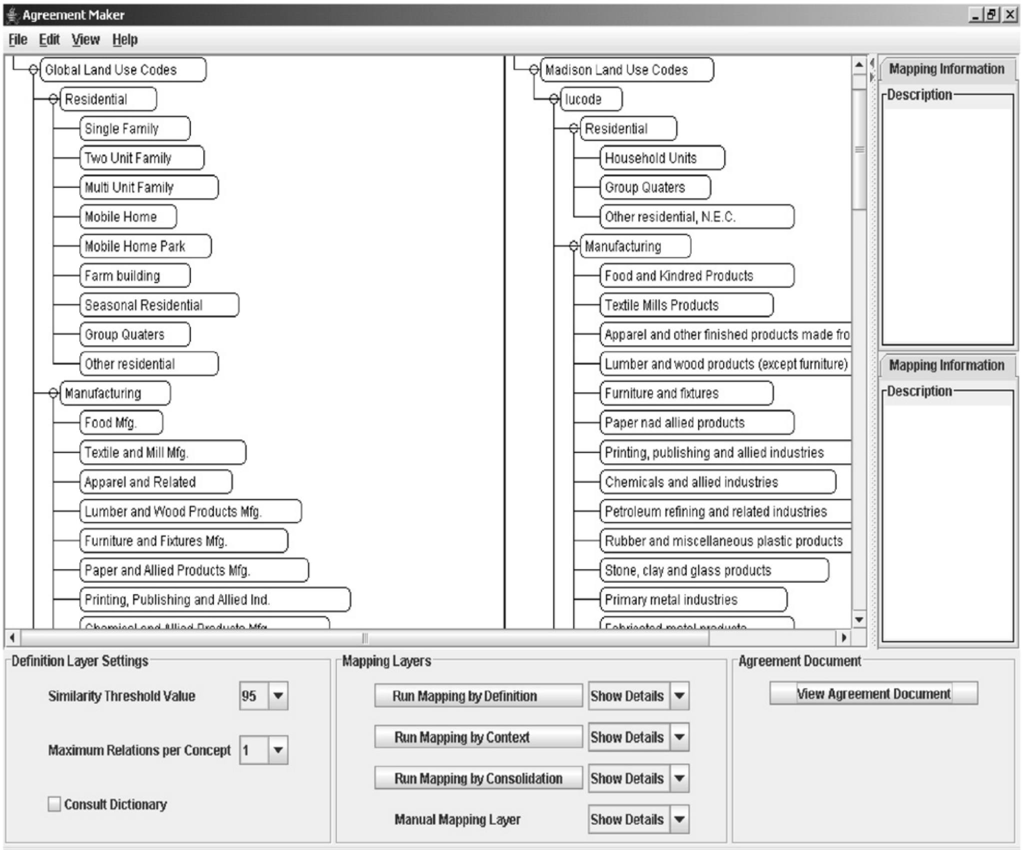
The *Agreement maker* is a visual software tool that is used to create the mappings between the global (source) ontology and a local (target) ontology and generate an agreement document. Our tool implements the four mapping layers discussed in Section 4. The interface of our tool allows the user to load two ontologies side by side as shown in Fig. 6. The global (source) ontology is displayed on the left-hand side and the local (target) ontology is displayed on the right-hand side. Concepts names are displayed in rectangular nodes on the ontological trees.

5.1. *System architecture*

Our tool consists of four main modules: The *graphical user interface*, the *ontology parser*, the *mapping engine*, and the *agreement document generator*. Fig. 7 shows a diagram of the architecture of our tool.

5.1.1. *Graphical user interface*

The graphical user interface assists the user in making the mapping decisions. It is customizable, allowing the user to select colors for the various objects and to expand or collapse parts of the ontologies. The menu bar supports standard operations such as opening files, undoing and redoing actions, and getting help. The central part of the user interface displays the rendered ontologies. It also displays the results of the mappings that result from the various mapping layers. The description pane displays information such as comments, properties, and class relationships for any selected concept (we note that no



Definition Layer Settings

Similarity Threshold Value 95

Maximum Relations per Concept 1

☐ Consult Dictionary

Fig. 6. The graphical user interface of the AgreementMaker showing the menu bar (top), the description pane (right) and the control panel (bottom).

concept has been selected in Fig. 6). This information can help the user when mapping concepts manually. Finally, the control panel contains buttons to invoke the various automatic mapping procedures and to select whether to show or hide results of any of the four mapping layers. We give details of the user interaction in Section 5.2.

5.1.2. Ontology parser

The AgreementMaker maps ontologies expressed in XML, RDFS, OWL, or N3 [11]. These ontologies are parsed and converted into our own tree structures using this module. The ontology files are parsed using various application programming interfaces (APIs): Xalan APIs are used for XML schemas, Jena APIs [17] are used for RDFS [18] ontologies, and OWL Pellet [19] is used for N3 and OWL ontologies.

5.1.3. Mapping engine

This module is responsible for running the matching algorithms of the four mapping layers on the loaded ontologies. The mapping engine reports the matching results in the form of lines connecting concepts from the global (source) ontology to the local (target)

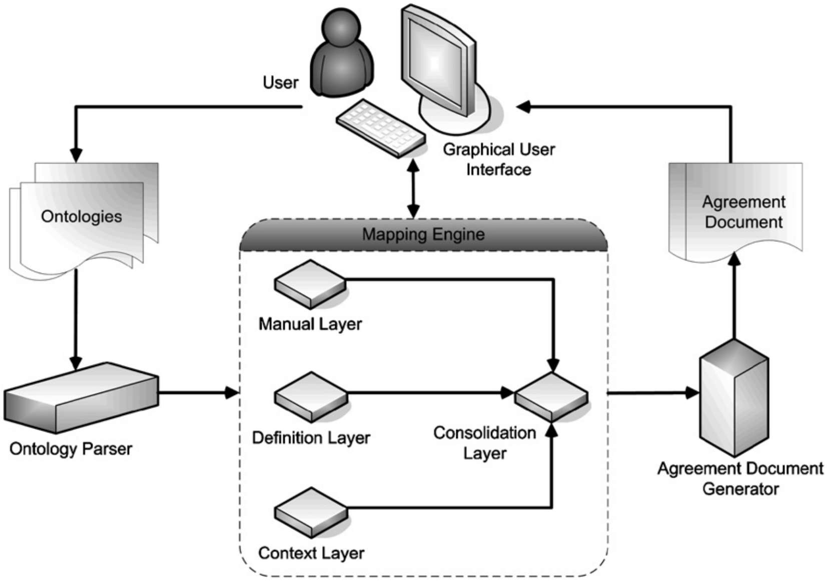


Fig. 7. System architecture of the *Agreement Maker*.

ontology. In a typical alignment session, the user invokes the definition mapping layer, then performs manual mappings, and then invokes the mapping by context layer in an iterative fashion. After the mappings are established using the three layers, the user invokes the mapping by consolidation layer to generate the final results.

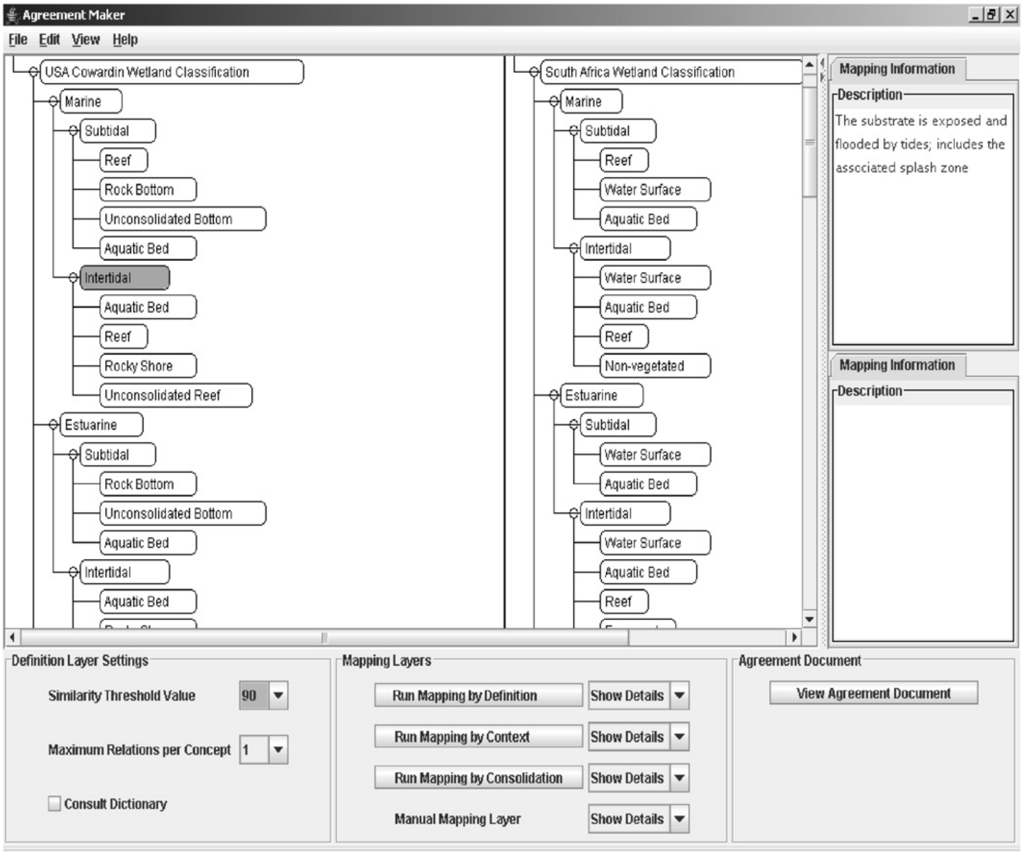
5.1.4. Agreement document generator

This module takes the information of the mappings generated by the mapping engine and stores it in the agreement document, which is the final output of the alignment process. The agreement document contains the alignment information that relates the two ontologies in an XML file. In addition, our tool is extendible and can be configured to reformat the agreement document in any format that is convenient for the end systems that use the document.

5.2. User interaction

Our tool enables the user to map concepts from the global (source) ontologies to local (target) ontologies using the graphical user interface. Upon loading the ontologies in our tool, they will be displayed in tree like structures as shown in Fig. 8. In the figure, the user loaded the Cowardin wetland classification as the source ontology, which appears on the left-hand side and the wetland classification system used in South Africa as the target ontology, which appears on the right-hand side.

In designing our user interface, we addressed many issues which affect the usability of our tool. We list several of these issues next and outline the rationale for our design decisions.



Description

The substrate is exposed and flooded by tides; includes the associated splash zone

Description

Definition Layer Settings

Similarity Threshold Value 90

Maximum Relations per Concept 1

☐ Consult Dictionary

Fig. 8. Description of the selected concept appears in the description pane.

5.2.1. *Ontology display*

Because the geospatial ontologies that we have been using are trees, we chose to display them as outline trees instead of using a general drawing tool such as GraphViz (www.graphviz.org). By using the outline tree paradigm for displaying trees, ontology browsing is similar to directory browsing, which is familiar to most users. We have implemented such trees so as to allow for the selective expansion or contraction of parts of the tree to facilitate browsing and mapping especially in the case when the ontologies are large.

5.2.2. *Meta information display*

Ontologies, such as those represented in OWL, have properties associated with their concepts, which can be used in the definition layer to compute similarity among the concepts. In addition, by displaying it, such information can be taken into account by the user when establishing manual mappings. If this information were displayed on the main canvas together with the ontologies, then it would cause visual overloading. Furthermore, it could interfere with the readability of the concept names. For these reasons, we provide a description pane (which can be hidden when not needed) that displays the description of any selected concept. For example, the description of the concept *Intertidal* appears in the upper part of the description pane as shown in Fig. 8. The upper part of the description pane is dedicated to the display of information that is associated with the concepts of the global (source) ontology while the lower part is dedicated to the display of information for the local (target) ontology.

5.2.3. *Similarity display*

Upon invoking the mapping by definition layer, a measure of the similarity among concepts is calculated to determine possible mappings and lines are drawn that display those possible mappings. To increase the clarity of the picture, the user can specify a similarity threshold so that only the lines that have similarity measures greater or equal to the selected threshold will be displayed. In addition, a maximum number of such possible mappings (as associated with each concept) can be specified. Fig. 9 shows the result of running the definition layer, as displayed to the user. In this particular case, the user has selected a threshold of 75%. The user also selected to see a maximum of two relations per concept in the source ontology. For example, the concept *Aquatic Bed* of the *Subtidal Estuarine* subcategory in the source ontology is related to the concept *Aquatic Bed* of the *Subtidal Marine* subcategory with a similarity measure of 75%. At the same time, it is related to the concept *Aquatic Bed* of the *Subtidal Estuarine* subcategory in the target ontology with a similarity measure of 100%. As previously mentioned, the user can collapse and expand the ontology trees to display only the concepts of interest and therefore only the associated similarities. To facilitate the reading of the lines that display the matched concepts, they can be highlighted and made bold when the associated concept is selected. For example, in Fig. 9 the concept *Aquatic Bed* has been selected and therefore both the matching lines that are connected to it and their similarity measures are highlighted and made bold.

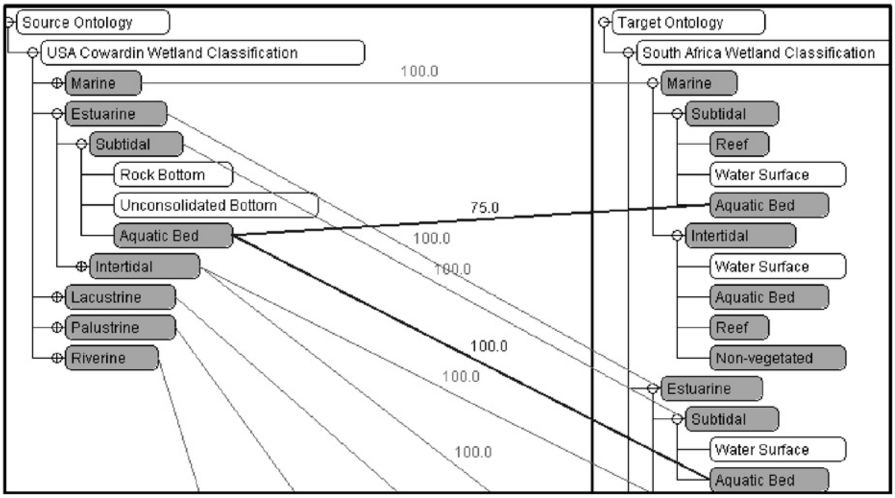


Fig. 9. Results of running the mapping by definition layer.

5.2.4. Manual mapping

To facilitate the manual mappings that are performed by the user, a mapping menu is used that contains all the mapping types. The user can select one or more concepts in the global (source) ontology and map them to one or more concepts in the local (target) ontology. Fig. 10 shows an example where the user is mapping the concept *Aviation* to the concept *Aircraft Transportation* as an exact match. After manually establishing a connection between those concepts, a menu that displays the different mapping types is displayed, so that the most appropriate mapping type can be chosen.

5.2.5. Context mapping

Upon invoking the mapping by context layer, more mappings between concepts may be discovered automatically as previously discussed in Section 4.3. For consistency, we display the result in a similar fashion to the results of the manual layer.

5.2.6. Simultaneous matching displays

An important issue is related to the presentation of the results of all the mapping layers to the user without visual overloading. To resolve this issue, the concepts that are mapped in a given layer are displayed using the same color. In this way, the user can easily distinguish which concepts are mapped by which layers. Another issue is the display of matching concepts that have been mapped by more than one layer. To improve readability, we allow the user to hide the results of any of the mapping layers and to redisplay them as desired. In this way, users can focus only on a subset of the layers at a time. Fig. 11 shows the mappings that result from three of the mapping layers. In the figure, lines with similarity measures result from the definition layer. Concepts that were connected in this way include *Personal Services*, *Business Services*, and *Professional Services*. Other concepts such as *Finance*, *Insurance*, *and Real Estate* in the global ontology were mapped manually by the user. The automatic mapping by context layer contributed to mapping the concept *Commercial* in the global ontology to the concept *Services* in the local ontology and has the label *Superset*.

5.2.7. Color selection

We give the user the flexibility of choosing the colors of the various visual components. This feature may be important for users who suffer from color blindness or other related

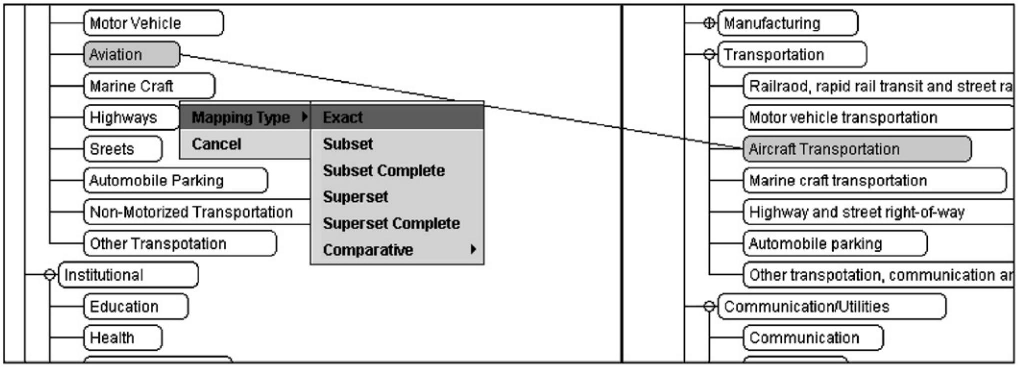
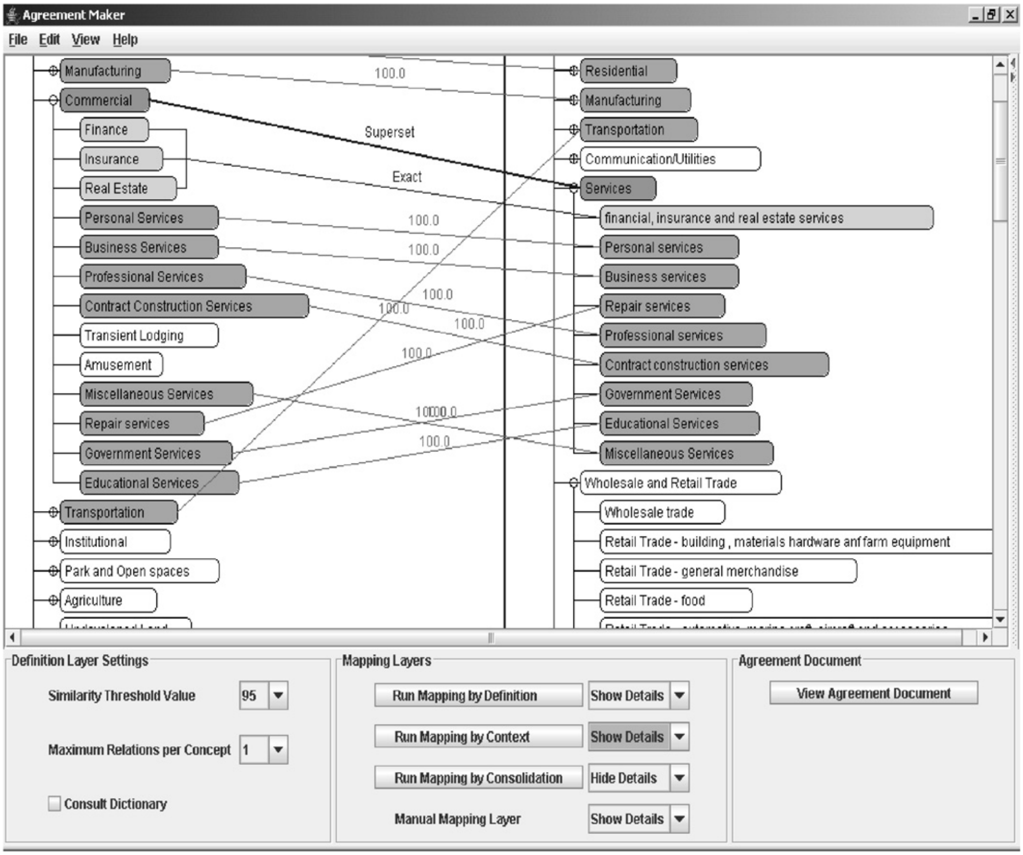


Fig. 10. Performing a manual mapping between two concepts.



Definition Layer Settings

Similarity Threshold Value 95

Maximum Relations per Concept 1

☐ Consult Dictionary

Run Mapping by Definition: Show Details

Run Mapping by Context: Show Details

Run Mapping by Consolidation: Hide Details

Manual Mapping Layer: Show Details

Fig. 11. Results of running three of the mapping layers.

visual problems. Our tool enables the user to change the color of the background, of the ontology, and of the highlighted concepts. The tool also enables the user to choose the color of the similarity lines that are produced by the definition mapping layer, by the manual mapping layer, and by the context mapping layer.

5.2.8. Mapping by consolidation

Once the mappings that are established by the definition mapping layer, by the manual mapping layer, and by the context mapping layer are finalized, the consolidation mapping layer can be invoked. After the application of this layer, not all the mappings are kept, as a result of the user's specified preferences. Also, the similarity measures of the definition layer are translated into the different mapping types that are used in the manual and context mapping layers, so as to show the semantics of the mappings. Furthermore, the lines that represent the mappings are color coded in a way that is consistent with the color associated with the layer that created them, for easy identification of their provenance.

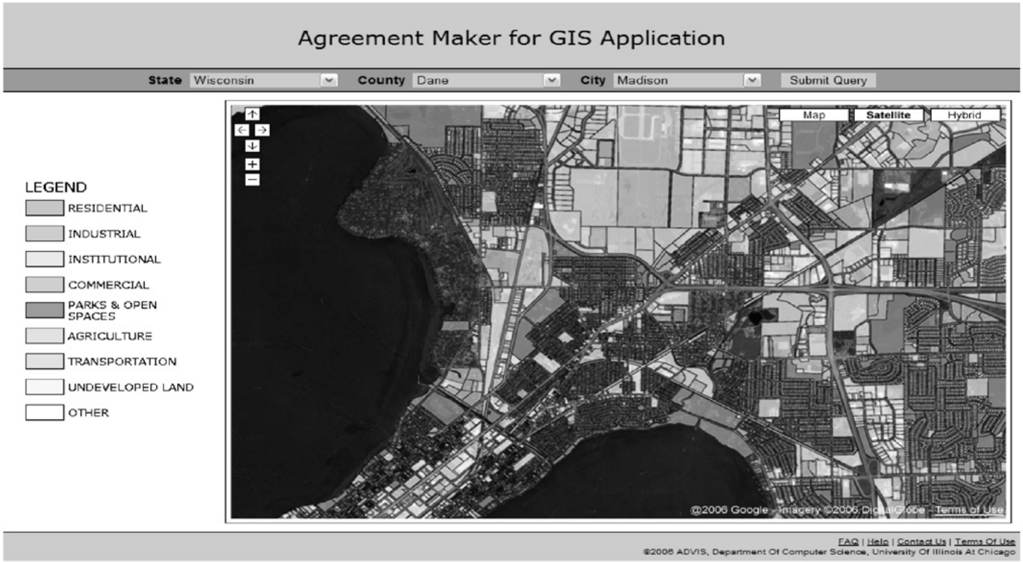
5.3. Agreement document

The result of the alignment process is a formatted document that contains all the mapping information. This document lists all concepts in the global (source) ontology in XML. Each global (source) concept has the following attributes:

- *ConceptID*: stores the name of the global (source) concept.
- *DefMapping*: stores the name of the concept in the local (target) ontology that has the highest similarity to the global (source) concept as determined by the automatic definition layer, if it exists.
- *DefSimilarity*: stores the mapping type corresponding to the highest similarity measure as determined by the automatic definition layer, if it exists.
- *ManualMapping*: stores the names of all the concepts in the local (target) ontology that are mapped to the global (source) concept in the manual mapping layer, if they exist.
- *ManualType*: stores the mapping types if mapped manually from the global (source) concept to the concepts in the local (target) ontology.
- *ContextMapping*: stores the names of all the concepts in the local (target) ontology that are mapped to the global (source) concept in the automatic mapping by context layer, if they exist.
- *ContextType*: stores the mapping types if mapped by context from the global (source) concept to the concepts in the local (target) ontology.
- *ConsolidationMapping*: stores the names of the concepts in the local (target) ontology that are mapped in the consolidation layer.
- *ConsolidationType*: stores the mapping types obtained by the consolidation process.

5.4. Land use maps

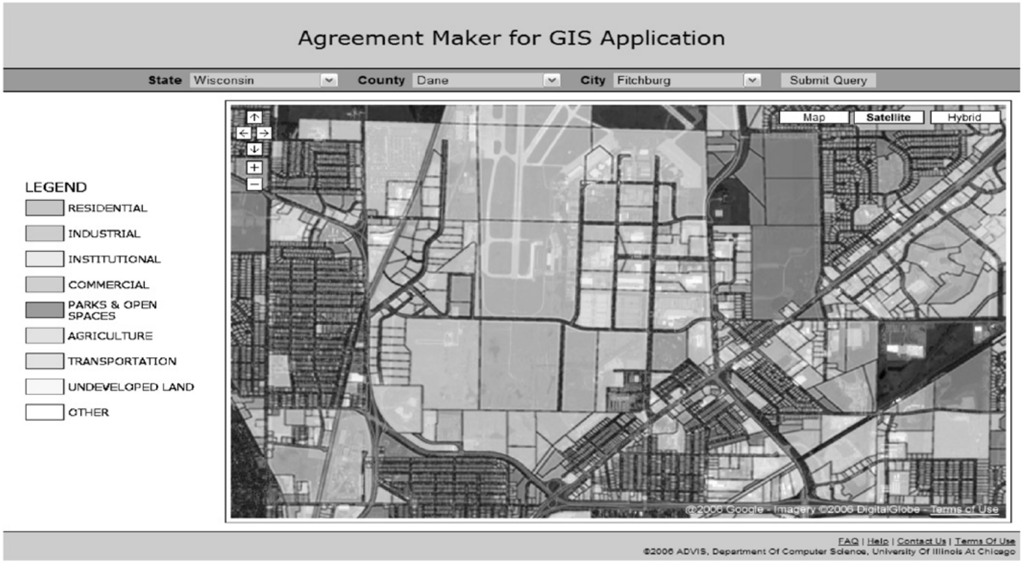
Having aligned the ontology describing the land use codes for the city of Madison and the one describing the land use codes for the Fitchburg township to a global ontology, we were able to produce the agreement documents for these two alignments. Agreement documents can be used to formulate a query to multiple local (target) databases using the concepts of a global (source) ontology. For example, a query that will ask for the *undeveloped land* in Madison and in the Fitchburg township, will return the land parcels that better match the concept of undeveloped land in the global ontology. Similarly, queries can be formulated for each of the other concepts of the global ontology. As a proof of concept, we have developed an interface to query and display all the land parcels in a specified region, such that the land parcels will be colored so as to indicate their land use code. Fig. 12 shows such an interface that depicts the land use map for the city of Madison. Similarly, the land use map of the Fitchburg township is depicted in Fig. 13. Our land use maps were superimposed on satellite images obtained from Google maps (http://maps.google.com). Land use maps serve as an example of geospatial interoperability where heterogeneities of the distributed databases are hidden from the user. In this way, an overall view can be obtained, which is not limited by the regional boundaries across which the data classification changes.


Agreement Maker for GIS Application

State Wisconsin County Dane City Madison

Submit Query
Map
Satellite
Hybrid
LEGEND
RESIDENTIAL
INDUSTRIAL
INSTITUTIONAL
COMMERCIAL
PARKS & OPEN SPACES
AGRICULTURE
TRANSPORTATION
UNDEVELOPED LAND
OTHER
@2006 Google - Imagery ©2006 DigitalGlobe - Terms of Use
FAQ | Help | Contact Us | Terms Of Use
©2006 ADVIS, Department Of Computer Science, University Of Illinois At Chicago

Fig. 12. Land use map for the city of Madison, Wisconsin.


Agreement Maker for GIS Application

State Wisconsin County Dane City Fitchburg

Submit Query
Map
Satellite
Hybrid
LEGEND
RESIDENTIAL
INDUSTRIAL
INSTITUTIONAL
COMMERCIAL
PARKS & OPEN SPACES
AGRICULTURE
TRANSPORTATION
UNDEVELOPED LAND
OTHER
@2006 Google - Imagery ©2006 DigitalGlobe - Terms of Use
FAQ | Help | Contact Us | Terms Of Use
©2006 ADVIS, Department Of Computer Science, University Of Illinois At Chicago

Fig. 13. Land use map for Fitchburg township, Wisconsin.

6. Validation of the multi-layered approach

Although this paper is essentially about the visual support that is provided by the AgreementMaker to align ontologies, it is important to determine the effectiveness of the overall alignment process, so as to validate the end result of our approach. By *effectiveness* we mean the percentage of correct mappings that are automatically determined as a percentage of the overall number of mappings that are needed to map two ontologies.

Until recently there was not a standard method to compare different alignment approaches. The ontology alignment evaluation initiative (OAEI) [4] intends to provide such a method of comparison. We note that the ontologies that are provided by this initiative are not geospatial and furthermore are not similar to our geospatial ontologies. Therefore, we proceeded in two ways: (1) we determined the effectiveness of our approach for our geospatial ontologies; (2) we determined the effectiveness of our approach for the ontologies that are provided by OAEI.

6.1. Geospatial ontologies

In the first case study of Section 3.1, we considered the alignment of the global ontology to the local ontology of the city of Madison and the alignment of the global ontology to the local ontology of the Fitchburg township. In the former case, 32% of the mappings were performed manually by the user while 60% of the mappings were performed automatically by the definition layer. The context layer performed 8% of the mappings. Finally, the consolidation layer generated the final mappings for all the concepts mapped in the previous layers. In the latter case, 46% of the mappings were performed manually by the user while 44% of the mappings were performed automatically by the definition layer. The context layer performed 10% of the mappings. Finally, the consolidation layer generated the final mappings for all the concepts mapped in the previous layers.

In the alignment of the two wetland ontologies that were presented in Section 3.2, 50% of the mappings were performed manually by the user while 50% of the mappings were performed automatically by the definition layer. The context layer performed 15% of the mappings, thus overlapping with the mappings of the definition layer but without any mapping conflicts. Finally, the consolidation layer generated the final mappings for all the concepts mapped in the previous layers picking the overlapping 15% mappings from the layer that was specified first by the user in the priority sequence.

Fig. 14 shows a chart summarizing the results of our three mappings in the geospatial domain. It puts in evidence that most of the mappings are performed by the automatic definition layer.

6.2. OAEI ontologies

To further validate our approach, we aligned three sets of ontologies provided by the OAEI [4]. The first set contained two ontologies describing attributes of people and pets, the second set contained two ontologies describing classifications of various weapon types, and the third set contained two ontologies describing classifications of computer networks and equipments. In the first set, 87% of the mappings were performed automatically: 71 concepts out of a total 81 concepts were successfully mapped by the automatic definition layer with high similarity measures, while only 11 were mapped manually by the user. In the second set, 95% of the mappings were automatically performed: 74 concepts out of a total of 78 concepts were successfully mapped by the automatic definition layer with high similarity measures, while only four were mapped manually. In the third set, 52% of the mappings were automatic: 11 concepts out of a total 27 concepts were successfully mapped

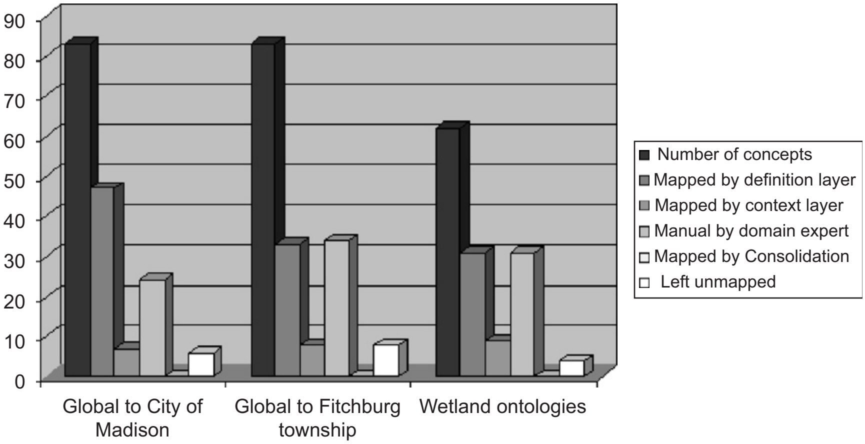


Fig. 14. Summary of the results obtained for the three alignments in the geospatial domain.

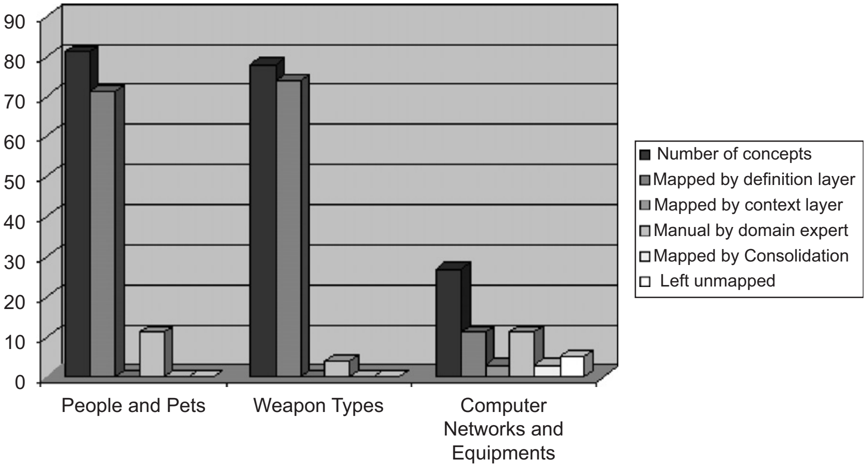


Fig. 15. Summary of the results obtained for the alignments of the three sets of OAEI ontologies.

by the automatic definition layer with high similarity measures, three concepts were successfully mapped by the automatic context layer, 11 concepts were mapped manually, and the mappings of three concepts were resolved in the consolidation layer. Fig. 15 shows a chart summarizing the results of mapping the three sets.

6.3. Discussion

The distribution of the mappings by the four layers is dependent on the type of the ontologies being aligned. If the concepts of the ontology tree are very close in definition and meaning, then the definition mapping layer will most likely yield a high percentage of mappings. The context mapping layer usually yields a high percentage of mappings when the ontological trees are structurally similar and the mapping of the lower level concepts can be performed by the definition mapping layer (or by the user). In this case, the deduction process yields a high percentage of automatically performed mappings. It was also found that the deeper the ontology tree the greater the potential for mappings that can be performed by the context mapping layer, as compared with the total number of other types of mappings. This stems from the fact that once the leaves are mapped then the deduction process can take over without the need for other types of mappings [2]. When conflicts occur, the consolidation mapping layer is the most valuable. Conflicts may occur when different mappings are arrived at by the different layers. In particular, manually established mappings may contradict mappings such as those established by the context mapping layer [3].

A comparison between the results that were obtained for the geospatial ontologies and for the OAEI ontologies shows that the effectiveness of the different layers is strongly dependent on the actual ontologies under consideration. Our geospatial ontologies contain short and sparse definitions of the different terms. Therefore, the automatic mapping by definition layer can only play a limited role, whereas the emphasis of OAEI ontologies appears to be on testing the effectiveness of the definition mappings. The fact that we have components in our alignment process that address different kinds of similarities is, therefore, one of the strengths of our approach. We expect that in the future different types of OAEI ontologies will be proposed that will test alignment strategies more thoroughly.

7. Conclusions

In this paper, we described our multi-layered approach to data integration in the geospatial domain. To demonstrate our approach, we presented two case studies, illustrating two architecture types, for land use codes and for wetlands. Our multi-layered alignment approach consists of four layers of mappings. In the first layer, the concepts of the ontologies are automatically mapped to one another based on their definition as provided by a dictionary or by information associated with the ontology concepts. In the second layer, the user manually maps concepts. In the third layer, an automatic procedure is invoked which deduces more mappings by looking at previous ones and by considering the topology of the ontology. Finally, in the fourth layer, an automatic procedure is used to resolve potential conflicts by attributing preferences to the results provided by the different layers.

We have developed the AgreementMaker, an alignment tool that implements the aforementioned four layers and produces an agreement document that contains the final mappings among concepts in the two ontologies being aligned. One of the most important components of the AgreementMaker is its graphical user interface. Many issues were considered in the design of the graphical user interface of the AgreementMaker. In this paper we describe in detail how the user interacts with the interface and the various design choices that were made so as to support all the steps that are necessary for the alignment of ontologies.

As a proof of concept, we have shown how the information contained in an agreement document can be used to enable querying in a distributed and heterogeneous geospatial application for land use codes. In our example, the heterogeneities of the distributed land use databases are hidden from the user who needs only to formulate queries in terms of a global ontology. To validate our multi-layered approach, we studied its effectiveness in several case studies, which include both geospatial and non-geospatial examples.

In the future, we plan to enhance both the mapping algorithms and the graphical user interface. For the mapping algorithms, we would like to further explore the topology of the ontologies. User studies will be valuable in determining possible improvements to the graphical user interface. Furthermore, as the Ontology Alignment Evaluation Initiative (OAEI) progresses we will conduct more evaluation studies as well as comparisons with related tools. Finally, so as to deploy a distributed system that is fully extensible and scalable, we will investigate a middleware architecture for our integration framework.

Acknowledgments

We would like to thank Nancy Wiegand and Steve Ventura, from the Land Information & Computer Graphics Facility at the University of Wisconsin-Madison for the discussions on land use problems. We would like to also thank them for providing us with land use code information for several cities and counties in the state of Wisconsin.

References

[1] I.F. Cruz, A. Rajendran, W. Sunna, N. Wiegand, Handling semantic heterogeneities using declarative agreements, in: Tenth International ACM GIS Symposium, 2002, pp. 168–174.

[2] I.F. Cruz, W. Sunna, K. Ayloo, Concept level matching of geospatial ontologies, in: GISPlanet Second Conference and Exhibition on Geographic Information, Estoril, Portugal, 2005.

[3] I.F. Cruz, W. Sunna, A. Chaudhry, Semi-automatic ontology alignment for geospatial data integration, in: Third International Conference on Geographic Information Science (GIScience), Lecture Notes in Computer Science, vol. 3234, Springer, Berlin, 2004, pp. 51–66.

[4] P. Shvaiko, J. Euzenat, A survey of schema-based matching approaches, in: Journal Data Semantics IV, Lecture Notes in Computer Science, vol. 3730, Springer, Berlin, 2005, pp. 146–171.

[5] E. Rahm, P.A. Bernstein, A survey of approaches to automatic schema matching, VLDB Journal 10 (4) (2001) 334–350.

[6] F.T. Fonseca, M.J. Egenhofer, C.A. Davis, G. Câmara, Semantic granularity in ontology-driven geographic information systems, Annals of Mathematics and Artificial Intelligence 36 (1–2) (2002) 121–151.

[7] D.L. McGuinness, R. Fikes, J. Rice, S. Wilder, An environment for merging and testing large ontologies, in: Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR 2000), 2000, pp. 483–493.

[8] D. Aumueller, H.H. Do, S. Massmann, E. Rahm, Schema and ontology matching with COMA++, in: ACM SIGMOD International Conference on Management of Data, 2005, pp. 906–908.

[9] N. Jian, W. Hu, G. Cheng, Y. Qu, FalconAO: aligning ontologies with Falcon, in: K-CAP 2005 Workshop on Integrating Ontologies, CEUR Workshop Proceedings 156, 2005.

[10] M.A. Hernández, R.J. Miller, L.M. Haas, Clio: a semi-automatic tool for schema mapping, in: ACM SIGMOD International Conference on Management of Data, 2001, p. 607.

[11] T. Berners-Lee, D. Connolly, E. Prud'homeaux, Y. Scharf, Experience with N3 rules, in: W3C Workshop on Rule Languages for Interoperability, 2005.

[12] Y. An, A. Borgida, J. Mylopoulos, Inferring complex semantic mappings between relational tables and ontologies from simple correspondences, in: OTM Confederated International Conferences (Part II), Lecture Notes in Computer Science, vol. 3761, Springer, Berlin, 2005, pp. 1152–1169.

[13] J.H. Gennari, M.A. Musen, R.W. Fergerson, W.E. Grosso, M. Crubézy, H. Eriksson, N.F. Noy, S.W. Tu, The evolution of Protégé: an environment for knowledge-based systems development, International Journal of Human–Computer Studies 58 (1) (2003) 89–123.

[14] L.M. Cowardin, V. Carter, F.C. Golet, E.T. LaRoe, Classification of wetlands and deepwater habitats of the United States. US Department of the Interior, Fish and Wildlife Service, Washington, DC, Jamestown, ND: Northern Prairie Wildlife Research Center Online. ⟨http://www.npwrc.usgs.gov/resource/wetlands/classwet/index. htm⟩ (Version 04DEC1998), 1979.

[15] J. Dini, G. Gowan, P. Goodman, South African National Wetland Inventory, Proposed Wetland Classification System for South Africa, ⟨http://www.ngo.grida.no/soesa/nsoer/resource/wetland/inventory_classif.htm⟩, 1998.

[16] E. Hovy, Combining and standardizing large-scale, practical ontologies for machine translation and other uses, in: A. Rubio, N. Gallardo, R. Castro, A. Tejada (Eds.), First International Conference on Languages Resources and Evaluation (LREC), 1998, pp. 535–542.

[17] B. McBride, Jena: a semantic web toolkit, IEEE Internet Computing 6 (6) (2002) 55–59.

[18] B. McBride, The resource description framework (RDF) and its vocabulary description language RDFS, in: S. Staab, R. Studer (Eds.), Handbook on Ontologies, Springer, Berlin, 2004, pp. 51–66.

[19] G. Antoniou, F. van Harmelen, Web ontology language: OWL, in: S. Staab, R. Studer (Eds.), Handbook on Ontologies, Springer, Berlin, 2004, pp. 67–92.