

Matching Schemas for Geographical Information Systems Using Semantic Information

Christoph Quix, Lemonia Ragia, Linlin Cai, and Tian Gan

Informatik V, RWTH Aachen University, Germany
{quix, ragia, cai, tian}@i5.informatik.rwth-aachen.de
<http://www-i5.informatik.rwth-aachen.de/>

Abstract. Integration and interoperability is a basic requirement for geographic information systems (GIS). The web provides access to geographic data in several ways: on the one hand, web-based interactive GIS applications provide maps and routing information to end users; on the other hand, the data of some GIS can be accessed in a programmatic way using a web service. Thereby, the data is made available for other GIS applications. However, integrating data from various sources is a tedious task which requires the mapping of the involved schemas as a first step. Schema matching analyzes and identifies similarities of two schemas, but all approaches can be only semi-automatic as human intervention is required to verify the result of a schema matching algorithm. In this paper, we present an approach that improves the matching result of existing solutions by using semantic information provided by the context of the geographic application. This reduces the effort for manually correcting the results which has been validated in several application examples.

1 Introduction

Integration and interoperability is a basic requirement for geographic information systems (GIS). The existence of a variety of geographical data is very important for geographic applications. A lot of phenomena can be explained using the knowledge extracted from data of past years. However, integrating data from various sources is a tedious task which requires the mapping of the involved schemas as a first step. There is sometimes insufficient description of the data, with no clear explanation of the entities, attributes etc. Standards like GML (Geography Markup Language, <http://www.opengeospatial.org/>) have been proposed but are not yet widely in use.

Since geographic data tends to be collected from various sources and archived locally, most geographic databases are heterogeneous, i.e., different types, different resolutions and different spatial properties under different formats. The problems that might arise due to *heterogeneity* of the data include *structural heterogeneity* (*schematic heterogeneity*) and *semantic heterogeneity* (*data heterogeneity*) [10]. The semantic conflicts occur when semantically similar information is represented by, for example, different data structures in different local databases.

Research in *schema matching* aims at providing automatic methods to identify relationships between different schemas [14]. The output of a schema matching system should be a mapping that enables the translation of data from one database to

another database. Such a mapping can be expressed in form of a query which extracts the data from the source and transforms it into the schema of the target. Many aspects have to be considered during the process of matching, such as data values, element names, constraint information, structure information, domain knowledge, cardinality relationships, and so on. Several approaches have been proposed for schema matching [18,20]. There exist also a few matching prototypes such as Protoplasm [2] and Coma(++) [1,4]. They use mainly linguistic and structural schema matching techniques, sometimes in combination with external information such as thesauri to identify synonyms or similar terms.

Related to schema matching is ontology matching or ontology alignment [6]. An *ontology* provides a shared and common understanding of a domain that can be communicated between people with distributed or heterogeneous application systems [11]. In contrast to schemas which mainly describe the structure of data, ontologies rather define the semantics of data. While semantically rich modeling languages (such as EER or UML) are used at design time, the semantics is usually not becoming part of the database implementation (one reason might be performance). Thus, this semantic information is not available for schema matching.

We have made the experience that application of schema algorithms requires a detailed understanding of these algorithms to find the “right” algorithm and the “best” parameters. It also requires expertise in the domain of the schemas to be matched because no algorithm can deliver a perfect result; manual verification of the proposed mappings is always required. As this can become a tedious task for large schemas, our idea is to apply the knowledge that has been formalized as ontologies for a domain to the problem of schema matching. In particular, we use ontologies to identify incorrect mappings in a result of schema matching algorithm. In this paper, we show how this methodology can be applied to support schema matching in GIS. We propose to extend our generic schema matching system with a specific component for matching of GIS schemas which exploits the semantic information given in ontologies, type hierarchies, or taxonomies to improve the schema matching result.

In the following, we will first discuss related work in section 2. Section 3 then presents the main ideas of our method. In section 4, we will discuss the application of our system to GIS schemas. Section 5 concludes our paper and points out future work.

2 Related Work

There have been many approaches for schema matching. The main reason for the various approaches is that each schema matching problem has its own characteristics and might require a specific solution. In the following, we focus on the approaches which are relevant for our work; surveys about schema matching are given in [18,20].

The *Cupid* algorithm [14] is intended to be generic across data models and has been applied to XML and relational examples. Schemas are represented as tree structures; the main idea of the algorithm is that the similarity of leaf nodes (attributes) contributes also to the similarity of inner nodes (elements/relations). The initial similarity values are computed by a linguistic method, which might also use auxiliary information such as thesauri. A similar idea is followed by the *Similarity Flooding* algorithm [15]. Schemas are represented as directed labeled graphs. Based

on the idea that if two nodes are similar then also their neighbors are similar, the similarity of two nodes in the graph is propagated to its neighbors. This procedure is repeated until a fix-point is reached. The initial input similarities can be computed by any kind of linguistic matching method. The algorithm can be applied to any kind of graph structure. The *COMA* schema matching system is a platform designed to combine multiple matchers in a flexible way [4]. It provides a large number of individual matchers, which contains both terminology approaches and structural approaches. *COMA* also allows users to reuse the previously obtained matching results. *COMA++* [1] is an update of *COMA* and supports also ontologies as inputs and provides several matchers for ontology matching.

An early work on schema matching in GIS [16] is based on structural data description, finding similarities between the attributes and entities using various criteria. The aim was to identify a single schema to be used in land-use information systems. A matching system based on machine learning is proposed in [13]. The goal is to construct a mediated schema to allow uniform querying of multiple sources. Another approach provides the integration of spatial databases using different scales [5]. It involves schema matching process finding inter-schemas correspondences which are based on instance level relationships. In [17], a schema integration method is proposed, that consists of two steps: in a first step, the relationships between the source schemas are identified. Then, an integrated schema is generated and mappings between the integrated schema and the source schemas are established.

Other approaches focus on the semantic integration of geographic ontologies. For example, a formal concept analysis to integrate several geographic domain ontologies to one top-level ontology is used in [12]. Similar concepts are identified if they have similar characteristics or properties. A methodology for comparing categories among geographic ontologies is presented in [8]. Another aspect is addressed in [7], in which the authors use ontologies for the design of the system. They provide a formal framework for expressing the mappings between the ontologies used by the domain specialist and the models used by the information systems engineer.

3 Semantic Schema Matching

A common problem of schema matching algorithms is that they produce only approximate results, i.e. the similarity values are only one indicator for the similarity of two schema elements. Therefore, advanced matching systems such as *COMA++* [1] and *Protoplasm* [2] follow a hybrid approach in which several matching techniques can be combined. However, also the combination of several matching algorithms cannot produce a result for which we can say with 100% confidence that two elements are or are not a match. Usually, one will consider only matching elements with a similarity value above a certain threshold. This might have the effect that too many matching elements (including incorrect matches) are detected if the threshold is too low. On the other hand, if the threshold is too high, only a few elements will be matched (and even this result might still contain incorrect matches). In any case, manual effort is necessary to verify the result, i.e. remove incorrect matches and insert matches that have not been detected.

Therefore, our approach tries to identify the wrongly matched elements to improve the result of a schema matching algorithm and thereby reducing the manual effort for

the verification of the result. The idea is to use a logical formalism to represent the schemas, their constraints and the relationships between them. Then, we use this logical representation to validate the detected mappings between the schemas.

The main problem of this idea is to have information that can be used to identify incorrect mappings. In our system, we use ontologies for that purpose. An ontology defines a set of concepts and properties, which are connected by different types of relationships (e.g. specialization, generalization, synonyms) for a specific domain. Recently, ontologies have been formalized and standardized in the semantic web area in form of the Web Ontology Language (OWL). In this context, ontologies are defined as a set of formulas in description logics which may contain constraints and rich semantic relationships (subClassOf, cardinality constraints, disjunction, etc.).

The use of ontologies has several advantages. Firstly, ontologies contain the semantic information in form of logical formulas which is required to identify incorrect mappings. Detecting inconsistencies in an ontology is a very common task in this domain. Secondly, ontologies have usually a broader context than schemas. Ontologies model the knowledge of a domain whereas a schema is often limited to a specific application. Therefore, ontologies can be used to bridge the “gap” between two schemas. Finally, there are currently many efforts to develop standardized ontologies (expressed in OWL) for several domains. As we will explain in section 4, we are currently implementing a component that can make use of the semantic information expressed in weaker formalisms such as taxonomies or type hierarchies.

In the following, we will first explain in more detail the basic idea and the individual steps of our methodology and then discuss implementation issues.

3.1 Methodology

The outline of our solution is sketched in Fig. 1. First, the schemas are matched with an ontology. This matching results in a mapping between the elements of the schemas and the concepts of the ontology. Then, the two schemas are matched. All the discovered mappings are then integrated into an extended ontology which will be checked for inconsistencies.

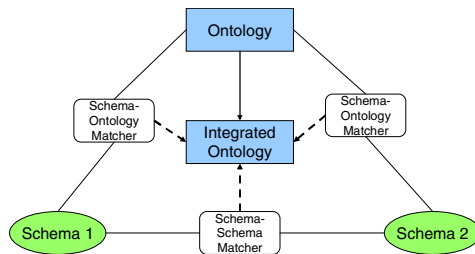


Fig. 1. Outline of the Semantic Schema Matching Algorithm

Step 1: Matching Schemas with the Ontology

The first step is to match the schemas with the ontology. This matching step requires high precision, because the result has high impact on the following steps. As the elements in schemas and ontologies are usually organized in different structures, the

elements will be matched using only linguistic information. We assume that the terms used in the ontology and the schemas are at least similar. If this is not the case, also auxiliary information such as dictionaries and thesauri could be used to find matches between schema and ontology elements. It is also possible to use general purpose ontologies such as SUMO.

The result of this step is a mapping between schema elements and the elements of the ontology. The schema elements are inserted into the ontology as artificial “concepts”, related by “equivalent” statements to the original ontology elements.

Step 2: Matching Schemas

To match the schemas, any kind of matching algorithm can be used. The result is a list of mappings between the elements of the two schemas. In our system, we can use a simplified version of Cupid [14] or and Similarity Flooding [15].

As mentioned before, one advantage of our approach is that we are able to identify incorrect mappings (if this can be derived from the semantic information that is available). Therefore, it is possible to lower the threshold for the initial matching methods so that more matching elements are found.

Step 3: Extend Ontology with Mappings

In the third step, the equivalences implied by the detected mappings are inserted into the ontology. As we have already inserted “artificial” elements representing the schema elements, we just need to relate these elements with some additional “equivalent” statements.

Fig. 2 shows a fragment of an extended ontology. The original ontology contains classes such as “Country”, “City”, and “Continent”. The corresponding elements of the schemas have been defined as “equivalentClasses” of the original classes (e.g. “Schema1Continent”, “Schema2City”); these mappings have been identified by the schema-ontology matcher.

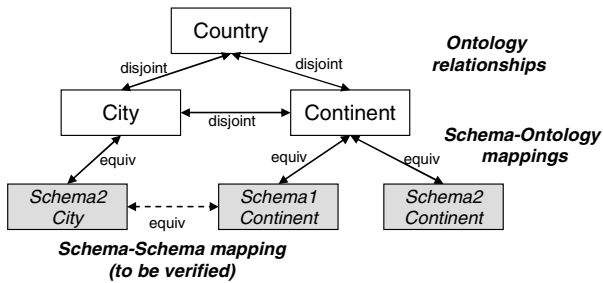


Fig. 2. Fragment of extended ontology

Step 4: Validate the Mappings

The extended ontology can now be sent to the reasoning system to detect the inconsistencies. This is done separately for each identified match. For example, the schema matching system has detected a match between the schema element describing the “name” of a city and the element describing the “name” of a continent. We would then insert into the ontology a statement saying that the corresponding

properties of the classes are equivalent. This implies that the classes representing the domains of these properties are not disjoint (unless they are empty). If the ontology contains a statement that the classes (or some other equivalent or super classes) are disjoint, then this leads to an inconsistency in the ontology. Therefore, we can derive that the current mapping is not correct.

The example of Fig. 2 does not contain any definition of properties; for reasons of simplicity, we have added an equivalence between the corresponding concepts (“Schema1Continent” and “Schema2City”). Now, it is easy to verify that this mapping implies an inconsistency as “Schema2City” as equivalent to “City” and “Schema1Continent” is equivalent to “Continent”, but “City” and “Continent” are disjoint concepts.

3.2 System Architecture and Implementation

Our prototype has been implemented using the Microsoft .NET Framework 1.1 and C# as programming language. The whole process can be controlled from a GUI component which also displays the mapping results. The core of the system is the matching framework Protoplasm [2], a library that simplifies the implementation of matching algorithms. It contains several data structures (graphs, matrices) and operators which are often used in schema matching algorithms. These operators include several lexical matchers (such as tokenizers, n-gram-matchers) and also methods to iterate and navigate over the graph structure. We have extended Protoplasm with an import operator for ontologies and a specific schema-ontology matcher. We use the Racer system [www.racer-systems.com] as reasoning server which supports reasoning on OWL ontologies. The ontologies to be verified are sent to the Racer server separately, if Racer reports an inconsistency, the proposed mapping will be discarded.

3.3 Evaluation

We evaluated our system with some schemas from different domains. The first example is MONDIAL [http://www.dbis.informatik.uni-goettingen.de/Mondial/], a data set of geographical web data sources with information about cities, countries, etc. The second example is taken from GeneX [http://www.ncgr.org/genex/] database, a

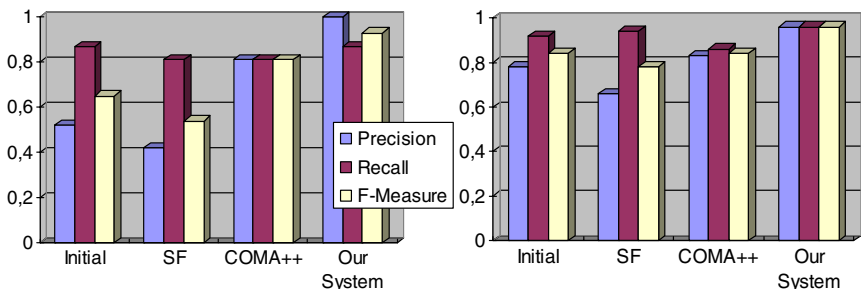


Fig. 3. Experimental Results for the MONDIAL (left) and GeneX (right) examples

database (and application) for the management of gene expressions. The ontologies for these examples have been created manually for an integration scenario, i.e., the ontologies were not tuned for the schema matching scenario.

The results of the evaluation are shown in Fig. 3, compared to Similarity Flooding (SF) and COMA++. In both cases, we used a simplified Cupid algorithm to produce initial schema matching result. As our system is only able to identify incorrect mappings and is not able to derive new mappings, only the precision of the result (the ratio between correct and incorrect mappings detected) could be improved compared to the initial results of the schema-schema matching algorithm. The performance of our system is comparable to other systems; the additional time for the semantic verification of the mappings is about 15-25% of the total time.

4 Semantic Schema Matching in Geographic Applications

4.1 Characteristics of Geographic Data

Geographic data include traditional geometric and thematic data. A typical GIS schema contains elements that (i) describe the geometry of the object, e.g. lines, shapes, polylines, etc.; (ii) describe other properties of the object (thematic data), e.g. surface of a road, population, data of the environment; and (iii) are used internally in the GIS such as identifiers, (foreign) keys. To achieve a high quality mapping between GIS schemas, we need to distinguish between these different types of elements. The idea is to apply specific matching techniques for each element type. Such a distinction fits nicely into our semantic approach, as specific ontologies for each element type can be applied separately.

Fig. 4 illustrates simple examples to show typical characteristics of of GIS schemas. The names of the elements (Road) are in this case the same, but there is the geometry of the objects does not match (Polygon vs. LineString). Although the names of the elements and some attributes are the same, the elements are characterized to be different. On the other hand, the similarity of the geometry might contribute to the overall similarity of the elements.

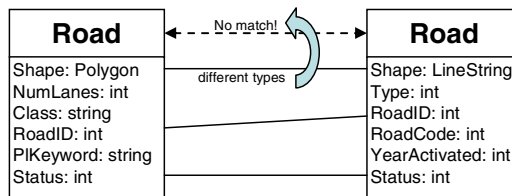


Fig. 4. Schemas with different geometry

4.2 Matching System for Geographic Information Systems

In order to bring the geometry in our matching approach, we make the extension in our system using the *Geography Markup Language* (GML). GML is an XML based encoding standard for geographic information developed by the *Open Geospatial*

Consortium (OGC, <http://www.opengeospatial.org/>). The main elements of the GML modeling structure are geographic features which represent real world geographical objects. Their geometric properties are modeled by a uniform geometric definition using two-dimensional coordinates. GML 2.0 is defined in XML Schema (XSD) and is divided into three parts: geometry schema, feature schema, and XLinks schema to provide support for linking the geographical data with other sources and to allow referencing on GML documents. The current version 3.1 of GML [3] also supports now also 3D geometry and complex values of features.

While the form of the geometrical description is defined very detailed, the form of the actual features is not. Instead GML offers a structural framework to fit the scenario it is meant to describe. Every model is made up of features. These features have properties. The properties have a name, a type and a value. The structure of the feature, hence the number and types of the properties, is defined by its type.

To identify semantic similarity of geometry types, we use the type hierarchy of GML geometry types [3]. For example, based on the hierarchy we can assume that 'Surface' and 'Polygon' are similar because they have the same direct super class. On the other hand, 'Polygon' and 'Curve' are not directly related in this hierarchy; thus, we can infer that these concepts are not similar. Of course, this is a very simplistic view of similarity of classes which has to be implemented using well-defined measures. Our implementation uses several different measures to compute the similarity of types (as defined in type hierarchies) or classes (as defined in ontologies). Type hierarchies and ontologies often do not contain the semantic information (explicit statements about disjointness or equivalence of concepts) which is required for our approach to detect logical inconsistencies. Therefore, we use the computed similarity values to enrich the ontologies with semantic information required. Examples for such measures are the linguistic similarity of class names and their synonyms, the properties of a class, and the similarity of semantically related classes. Such measures have been already defined in the context of ontology alignment [6,19]. More specific procedures for the geographic domain, that could be applied in this context, have also been described [8,12].

To analyze such relationships in geographic schemas, we have extended our schema matching system with *GeoMatcher* component. The *GeoMatcher* treats geometric information such as data types, value ranges and relationship types. It is also based on ontologies, and is therefore an extension of the ontological reasoning component of the core system. Please note that according to our discussion above, the similarity of geometry types might contribute to the similarity of the entities in which these geometry types are used. Because of this, the *GeoMatcher* might also add new mappings and not only delete mappings as the semantic matching component in the core system. A similar idea to derive indirect mappings has already been proposed earlier [21], but we use the knowledge contained in ontologies more explicitly.

The type hierarchy of GML is only one source for semantic information about geographic entities. In addition, we plan to use additional classifications and categorizations such as CORINE LC Classes [http://nfp-lv.eionet.eu.int/clc_db/en/classes/class_ndx.htm] or the ontology for the ISO standard Geographic Information – Metadata (ISO 19115:2003, <http://loki.cae.drexel.edu/~wbs/ontology/iso-19115.htm>). As discussed above, a problem here is that these ontologies often do not contain the required semantic information to detect logical inconsistencies. Therefore, a similarity

value for two classes is computed (using the measures described above) if no explicit information is given.

5 Conclusion

Integration of data is a task that is especially required for geographical information systems as certain phenomena can be only explained by looking at several data sources at the same time. Schema matching methods have been proposed in the recent years to support the task of information integration in a semi-automatic way. Still, schema matching requires manual intervention as the methods are not able to deliver perfect solutions.

In this paper, we have presented an approach that improves schema matching results by exploiting semantic information that has been formalized in ontologies. Ontologies represent the knowledge of a particular domain. By introducing this knowledge into the schema matching process, we reduce the manual effort that is required for schema matching significantly as we are now able to detect most of the incorrect mappings automatically. Therefore, an information systems engineer needs to spend less time on the verification of the results of a schema matching algorithm. Our approach can be used as an extension to any schema matching algorithm as it is independent of the previous steps; it just requires a list of mappings as input.

The application of our basic matching system to GIS schemas was also promising, but our analysis of geographic data has shown that more specialized matchers for GIS schemas are necessary. Therefore, we extended our matching system with a *GeoMatcher* component which takes into account the specific features of GIS schemas. As the ontologies in the GIS domain do not contain the required explicit knowledge about disjointness or equivalence of concepts, an important feature of the *GeoMatcher* is computation of similarity values of concepts based on the (limited) information given in the knowledge. An initial evaluation of the core part of the system showed very good results; however, more evaluation results in the area of GIS schemas are required to verify and optimize the *GeoMatcher* component.

Future work will concentrate on providing optimized matching methods for the different parts of a GIS schema. In particular, our focus is on using more semantic information for matching, even if it cannot be used to prove inconsistencies in a strict logical sense. We plan to use our generic metamodel *GeRoMe* [9] to ease the matching of models from different modeling languages (e.g., OWL, XML Schema, SQL). Using such techniques, we can provide further extensions and improvements for schema matching results which are not limited only to the GIS domain.

References

- [1] D. Aumüller, H.H. Do, S. Massmann, E. Rahm: Schema and Ontology Matching with COMA++. *Proc. Intl. Conf. on Management of Data (SIGMOD)*, 2005.
- [2] P.A. Bernstein, S. Melnik, M. Petropoulos, C. Quix: Industrial-strength schema matching. *SIGMOD Record*, 33(4):38-43, 2004.
- [3] S. Cox, P. Daisey, R. Lake, C. Portele, A. Whiteside: OpenGIS Geography Markup Language (GML) Implementation Specification. Version 3.1.0, 2004.

- [4] H.H.Do, E.Rahm. COMA: a system for flexible combination of schema matching approaches. *Proc. Conf. on Very Large Data Bases (VLDB)*, pp. 610–621, 2001.
- [5] T. Devogele, C. Parent, S. Spaccapietra: On Spatial Database Integration. *Intl. Journal of Geographical Information Science*, 12(4):335-352, 1998.
- [6] J. Euzenat (Ed.): State of the art in ontology alignment. *Deliverable 2.2.3, Knowledge Web Project*, <http://knowledgeweb.semanticweb.org/>, 2004.
- [7] F. Fonseca, C. Davis, G. Camara: Bridging ontologies and conceptual schemas in geographic information integration. *Geoinformatica*, 7(4):355–378, 2003.
- [8] M. Kavouras, M. Kokla, E. Tomai: Comparing categories among geographic ontologies. *Computers & Geosciences*, Vol. 31, no. 2, pp. 145-154, 2005.
- [9] D. Kensché, C. Quix, M.A. Chatti, M. Jarke: GeRoMe – A Generic Role Based Metamodel for Model Management. *Proc. 4th Intl. Conf. on Ontologies, DataBases, and Applications of Semantics (ODBASE)*, Agia Napa, Cyprus, 2005.
- [10] W. Kim, J. Seo: Classifying Schematic and Data Heterogeneity in Multi-database Systems. *IEEE Computer*, 24(12):12-18, 1991.
- [11] M. Klein, D. Fensel, F. Harmelen, I. Horrocks: The Relation between Ontology and Schema-languages: Translating OIL-specifications in XML-Schema. *Proc. ECAI Workshop on Applications of Ontologies and Problem-Solving Methods*, Berlin, 2000.
- [12] M. Kokla, M. Kavouras: Fusion of top-level and geographic domain ontologies based on context formation and complementarity. *Intl. Journal of Geographical Information Science*, 15(7):679-687, 2001.
- [13] S. Manoah, O. Boucelma, Y. Lassoued : Schema Matching in GIS. *Proc. 11th Intl. Conf. on Artificial Intelligence: Methodology, Systems and Applications (AIMSA)*, Varna, Bulgaria, 2004.
- [14] J. Madhavan, P.A. Bernstein, E. Rahm: Generic schema matching with Cupid. *Proc. Conf. on Very Large Data Bases (VLDB)*, pp. 49–58, Rome, Italy, 2001.
- [15] S.Melnik, H. Garcia-Molina, E. Rahm: Similarity Flooding: A Versatile Graph Matching Algorithm. *Proc. 18th International Conference on Data Engineering (ICDE)*, pp. 117-128, San Jose, CA, 2002.
- [16] L.T. Nyerges: Schema integration analysis for the development of GIS databases. *Intl. Journal of Geographical Information Systems*, 3(2):153-183, 1989.
- [17] J. Park: Schema Integration Methodology and Toolkit for Heterogeneous and Distributed Geographic Databases. *Working Paper, University of Minnesota*, <http://miscr.umn.edu/workingpapers/abstracts/0131.aspx>, 2001.
- [18] E. Rahm, P.A. Bernstein: A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, 2001.
- [19] M.A. Rodríguez, M.J. Egenhofer. Determining Semantic Similarity Among Entity Classes from Different Ontologies. *IEEE Transactions on Knowledge and Data Engineering* 15(2):442-456, 2003.
- [20] P. Shvaiko, J. Euzenat: A Survey of Schema-based Matching Approaches. *Journal on Data Semantics IV*, LNCS 3730, pp. 146-171, Springer, 2005.
- [21] L. Xu, D.W. Embley: Using domain ontologies to discover direct and indirect matches for schema elements. *Proc. Workshop on Semantic Integration at ISWC 2003*, Sanibel Island, FL, 2003.