Uncertainty in the Automation of Ontology Matching

Valerie Cross
Department of Computer Science and Systems Analysis, Miami UniversityOxford, OH, USA
crossv@muohio.edu

Abstract

The exchange of information between two agents over the Semantic Web requires a means of translating between the "vocabularies" of the agents. Much research has focused on the use of ontologies for specifying an agent's knowledge and for exchanging information between agents. Effective communication between agents using different ontologies, however, requires determining the semantic interoperability, i.e., the agreement between the two agents' ontologies. Ontology matching is essential for the process of merging or aligning ontologies and for effective communication between agents. This paper presents a survey of several proposals for ontology matching and develops a framework for the process of ontology comparison from several different levels and views. The role of fuzzy set theory in measuring the quality of the match between two ontologies is examined.

1. Introduction

The term ontology, according to Webster's dictionary, means a "particular theory about the nature of being or the kinds of existents." Although first used in the area of philosophy, the term ontology has been used by researchers in a variety of areas such as artificial intelligence (AI), information retrieval (IR), database theory, linguistics, and eCommerce. Numerous definitions now exist depending on one's perspective. One of the most common and simply stated definitions is that an ontology is a specification of a shared conceptualization [11]. An ontology specifies a shared vocabulary used to model a domain of interest. This vocabulary describes the type of objects and/or concepts that exist, their properties and relations. Standard relations such as *is-a, part-of,* and *instance-of* have predefined semantics. A concept hierarchy is an ontology without attributes and only with *is-a* relations between concepts.

Ontologies have been developed for many purposes [28]. In software systems, they provide reusability and information sharing. In IR the search operation may use an ontology as metadata to help direct the information retrieval to more relevant sources. The need for reliability in various systems promotes the use of ontologies for consistency checking. In the specification process, an ontology may be used to assist in identifying requirements for a system. Researchers in areas such e-Commerce or geographical information systems are developing global standardized ontologies. But most agree that it is not feasible for each discipline or community to standardize. Even if standardization obstacles such as differences in practices and complexity and security issues are overcome, dynamic and unpredictable interactions between applications will require dynamic mapping between their different ontologies.

The building and use of ontologies on the World-Wide Web has dramatically increased and has started to replace older means of exchanging data. World-Wide Web users have information easily and readily available by accessing web pages. Most of these pages, however, are only in human-readable format, and therefore, unusable by software agents. To overcome this problem, researchers have responded with the promise of the Semantic Web [2], where data has structure and ontologies describe the semantics of the data. Software agents using ontologies can better understand the meaning of the data and thus locate and integrate data from a wide variety of sources for diverse tasks.

The Semantic Web, by its decentralized nature, promotes a proliferation in the number of ontologies. Many describe similar domains, but with different terminologies. Others have overlapping domains. Semantic Web technology should foster knowledge exchange by providing tools to enable semantic interoperability [29]. Interoperability is established by discovering semantically appropriate mappings between different and independent ontologies.

A wide variety of methods have been proposed that (semi) automatically discover mappings between ontologies. The goal of this paper is to provide an overview of this research and to begin to investigate the role of similarity and aggregation in this process. Section 2 specifies a common meta-model for representing ontologies, examines issues of within ontology vs. between ontology matching, and categorizes various frames of reference for comparing ontologies. Section 3 overviews several approaches for performing ontology matching based on the level of matching. Section 4 focuses on the role of fuzzy set similarity measures and aggregation in this process. Section 5 presents conclusions and discusses future work.

0-7695-1997-0/03 $17.00 © 2003 IEEE

2. Perspectives on Ontology Comparison

Due to the WWW and the vision of the Semantic Web, vast amounts of information stored in various domain-specific databases and web sites are potentially accessible. But the need for semantic interoperability limits this accessibility. Much research has been focused on achieving semantic interoperability through semi-automated schema and ontology matching and integration. A survey of research on schema matching can be found in [22] while a survey of ontology-based integration can be found in [30]. Although they have been addressed separately, schema and ontology matching are very much related. Schema matching attempts to find semantic correspondences between elements of two schemas usually within a database context [22] while the ontology matching compares two ontologies and tries to find for each concept in one ontology the most similar concept in the other ontology [7]. This section summarizes some issues and views related to this matching process.

2.1. Ontology Representation

The methods of representing an ontology are diverse and depend on the required level of detail and logic. In practice, a thesaurus, a simple concept hierarchy, a semantic net, a frame system, or a logical model may represent an ontology. For example, WordNet , a terminological ontology, is a collection of categories organized by a partial order that is induced by inclusion [18]. A much more detailed ontology, Cyc [13] is an axiomatized ontology whose categories are distinguished by axioms and whose definitions are stated in logic.

Numerous languages for representing ontologies have been proposed. These languages differ not only in the expressiveness but also in the level of formality. Because the integration of and mapping between ontologies encoded in different languages is a difficult challenge [26], many researchers [20] investigating semantic mapping between ontologies, assume a common frame-based knowledge model designed to be compatible with OKBC [4]. This model serves as a generic knowledge representation compatible with many existing knowledge-representation systems.

The main components of an OKBC-compliant knowledge model are classes, slots (either for a relationship or an attribute in object-oriented terminology), facets and instances. A class is a collection of objects described by identical properties. Classes are organized into a taxonomy or a specialization and generalization hierarchy, also referred to as a subclass–superclass hierarchy. The superclass represents a generalization of its subclasses, the subclass, a specialization of its superclass. Slots are associated with each class and are inherited by the subclasses. Slots (aka properties) are named binary relations between a class and either another class or a primitive type (such as a string or a number). Facets constrain the values taken on by slots, for example, the minimum or maximum value of a slot. An actual member of a class is referred to as an instance of the class.

2.2. Intra vs. Inter Ontology Comparison

An earlier limitation placed on the ontology matching process was that the comparison of lexical entries, classes, and slots must occur within a single ontology. Different approaches have been used to satisfy the use of a single ontology [3,5],. concepts of two distinct ontologies have been mapped into a pre-defined single shared ontology. For many applications, however, forcing users to commit to a single ontology is not practical. Instead existing ontologies are integrated into one shared ontology [1]. With a single ontology, the semantic similarity between components from the separate ontologies can be determined as a function of the path distance between terms in the one hierarchical structure [3].

Another semantic similarity measure within a single ontology is based on information content and uses the degree of informativeness of the immediate superconcept that subsumes the two concepts being compared [23]. More recently research in ontology matching [6,14, 19, 23] has focused on the dynamic environment of the Semantic Web which makes an a priori shared ontology impractical. This environment requires ontology matching to occur on different independent ontologies without forcing integration between the autonomous ontologies.

2.3. Categories of Ontology Comparison

The primary goal of ontology matching is to determine a correspondence or mapping between the two ontologies. This mapping function is also referred to as a match function [22]. Because ontologies can be compared from many different perspectives, numerous techniques for ontology matching exist and have been categorized based on their differences. A schema (intensional) based match differs from an instance-based (extensional) match in that it examines the ontology descriptions and not the actual data associated with instances of the ontology.

A schema-based match can be further categorized as at the element level if it provides a mapping among single elements or at the structure level if it uses groups of elements and their structure to find a match. To match between the simple elements like class or slot names, mappings are constructed based on IR techniques like tokenization and stemming [8] and the use of external aids like a thesauri for looking up synonyms. To further verify matching of slots, constraint based matching compares the values of respective slot facets such as data range or data type to determine agreement between the slots.

This paper focuses on schema-based ontology matching techniques proposed for OKBC-compliant independent ontologies with no overlying shared or integrated single ontology. Both element and structural level individual matching techniques and their combination are examined. Although systems using instance-based matching have been developed such as GLUE [7], this approach is not as practical since most current Semantic Web ontologies do not contain a significant number of instances for matching.

3. The Matching Process

Two important aspects of ontologies are their syntax and semantics. The syntax involves the specification of the legal lexicalizations of an ontology, i.e., the vocabulary of the ontology. The semantics specifies how the vocabulary is used to convey meaning, i.e., what objects exist, their attributes, what relations exist between the objects and so on. Two levels of granularity in ontology matching are the *element-level* and the *structure-level* [22].

Element-level matching techniques compute a mapping between individual terms used to label an element of the ontology such as class, slot, or facet. This level corresponds to the syntax of the ontology. Structure-level techniques compute a mapping between composite groupings or subgraphs within the ontology and corresponds to the semantics of the ontology [16]. In the following discussion these levels are used as a framework to provide an examples of ontology matching techniques.

3.1. Element-level Matching

When comparing two ontologies at the element level, the objective is to find for each element in first ontology its matching element in the second ontology. It is typical to determine a normalized value in [0, 1] that specifies the degree of similarity between the two elements. Two primary ways of determining similarity at the element level are name matching and value matching.

3.1.1. Linguistic Name Matching. Often before name matching can begin, numerous techniques created by IR research need to be applied for preprocessing terms or names. For example, in [9] capitalization-based separation ("hireDate" becomes "hire Date"), same case conversion ("hire Date" becomes "hire date"), elimination of noise characters ("*bonus" becomes "bonus"), deletion of hyphens, and removal of stop words are used to greatly improve the performance of name matching. Once preprocessing is completed, name matching is performed in two different ways based on viewing the name as a set of words or as a single string.

Names may consist of multiple words. The word matching similarity for name n_1 and n_2 is calculated as the ratio between the number of common words in n_1 and n_2 and the total number of different words in n_1 and n_2. Words are determined to be common if they have identical spelling, sound the same based on an encoding, or have synonym matching using a thesaurus.

With string matching, each name is transformed by concatenating the words in the name into one long string. String similarity is then computed as the ratio between the length of the maximum common substring and the length of the longer string. For example, the names **student information** and **school student info** result in **studentinformation** and **schoolstudentinfo** with a common substring of studentinfo and a string matching similarity of 11/18. The similarities of the word set match and single string match are combined as a weighted average to produce an overall name similarity measure. These weights are user modifiable.

In [16], the Levenshtein edit distance [14] is used to measure the difference between two strings. It counts the minimum number of operations, i.e., insertions, deletions and substitutions, needed to change one string into another string. This edit distance is then converted to a string matching similarity measure.

3.1.2.Constraint-based Value Matching. Facets for slots often contain constraints, for example, to define data types and value ranges for a slot. If both ontologies specify such constraints, they can be used in the matching process to determine the similarity of slots [22]. For example, the similarity measure between two slots can factor in the matching of the values for their respective data types and their respective range facets.

In [16], the measure TSO (template slot overlap) is an example of constraint-based value matching based on the geometric mean value of how similar one slot's domain and range concepts (classes) are with another slot's. The geometric mean is used since a value converging to 0 is desired if either domain or range concepts completely fail to match. Notice that although this is an element level matching of slots, the evaluation requires using the similarity measures for structural matching of the concepts. This similarity measure is explained in the structural-level matching section.

Value matching in [9] looks at both the data type and the legal values for two slots (i.e., html fields). Borrowing from Cupid [15], the method uses a table to provide a similarity in [0, 1] between different predefined data types The similarity between two sets of legal values is the ratio of the intersection over the union of the two sets. The value match between two slots is then determined as a weighted average of the two components.

In both of the approaches, value-matching similarity for the two slots is combined in a weighted average with the linguistic name matching similarity value to produce an overall similarity measure between the slots.

3.2. Structure-level Matching

Structural-level matching compare combinations of elements that appear together in structure, i.e., two composite elements are being compared for similarity of their structure. The structure might be specified in different ways but often it is represented as a graph. Structure-level techniques, therefore, often are specified as graph matching algorithms that analyze all labels of nodes and arcs that are relevant (i.e., connected) to the element to be matched.

In [9] structure-level matching is referred to as composition matching and uses a graph-based matching algorithm. It assumes that the two elements, *u* and *v*, being matched are the pivot elements in their own subgraphs (representing structure besides that of the ontology) and then compares siblings of the two and the ancestors of the two. It uses name matching and value matching between the all pairs of ancestors and all pairs of siblings and produces a best match for siblings and a best match for ancestors. The overall structural similarity measure between u and v is determined as the weighted average of the sibling similarity and the ancestor similarity.

The semantic-level comparison in [16] compares semantic structures of ontologies based only on the taxonomy structure of the ontology. It is similar to the approach in [9] only at the higher level of the ontology itself. The similarity between two concepts in the ontology, referred to as the concept match is determined only based on the ancestors of the two concepts. It is calculated as the ratio of the intersection over the union of the two sets of ancestors. It is not clear how the intersection is determined but it probably is based on equality of linguistic names for ancestor concepts.

Notice that in both approaches, ancestor nodes in the graph are considered as contributing to the similarity of the two pivotal elements being compared. In [24] a slightly different approach is taken when performing structural-level matching between entity classes in two different ontologies. A semantic neighborhood of path distance *d* is defined for an entity class as all the entity classes reachable from the given entity class on a path of less than or equal to *d* undirected arcs. This approach leaves the meaning of the arcs open, i.e., it could be an is-a or a part-of arc, etc but restricted to relationship arcs. The use of undirected arcs means that both ancestors and descendents are considered in this structure-level matching. Intersection over semantic neighborhoods is approximated by the element-level matching of entity classes across the neighborhoods of the two ontologies.

Element-level matching between classes is slightly ambiguous, however, since it is based on measuring the similarity between associated synonym sets for each entity class and the similarity between the attribute sets for each entity class. The overall synonym (attribute) similarity for the two entity classes is assessed as the ratio of the intersection of their respective synonym (attribute) sets over the union of their respective synonym (attribute) sets. Then the overall entity class similarity is determined as a weighted average of the synonym and attribute similarities.

The use of ancestors and descendents similarity to contribute to the similarity of two pivot elements has been widely used and has been referred to as the Neighborhood Constraint [7]. The Glue system takes this one step farther and develops heuristic knowledge in the form of rules. For example, all other things being equal, the higher the value the percentage of matching children), the higher the probability of the pivotal elements matching.

Another recent approach to graph matching is the idea of similarity flooding [17], a hybrid matching algorithm that propagates similarity through the graph. Mapping between the nodes of the input graph are obtained by processing the graphs in an iterative fix-point computation. An initial mapping is obtained by syntactic string comparison of the vertices' names. The mapping is further specified within the fix-point computation.

4. Uncertainty in the Matching Process

Because the syntactic representation of the ontologies cannot completely describe the semantics of different ontologies, automatic matching of ontologies brings with it a degree of uncertainty [19]. For example, if only syntactic or element-level matching is performed, as is the case for name matching without the use of a thesaurus, inaccuracies can occur. Since name matching assumes similar names for attributes implies similar attributes, errors in matching may occur due to synonyms or homonyms. Thus each matching that is done has an associated degree of confidence which many systems specify by the degree of similarity.

Often the similarity measure is determined as a result of aggregating two or more similarity measures. For example, in [9] name matching is based on a weighted average of the string matching and the word matching similarity measures. Besides uncertainty in the similarity matching method, ontology matching and integration tools are starting to provide heuristic knowledge in the form of rules that also have uncertainty associated with them, for example, “two nodes match if their parents match and some of their descendants also match”[7].

Ontology matching and integration systems are performing a form of automated reasoning. A very useful tool would be the provision of a measure of accuracy that allows a user to determine his own tolerance to imprecision in the matching process. Many systems have a threshold level that if not met, then ignore the possible matching of elements between the ontologies. Another approach provides rules for using the threshold levels where in some cases, the system could be instructed to request help either if a threshold has not been satisfied or if the imprecision in the matching process becomes too great.

Some systems such as GLUE [7] provide a more generic form of matching that permits flexibility in selecting the similarity measure. Many of the similarity measures presented in various ontology matching systems are based on set theoretic measures of similarity [27]. The various forms of fuzzy set similarity measures and aggregation operators could be useful in the computation of the individual element-level similarity and the structure-level similarity. In a dynamic environment such as the envisioned Semantic Web agents themselves might like to specify their own similarity measures and aggregation operators in determining the semantic similarity between two autonomous ontologies.

5. Future Work

This initial survey of the ontology matching serves as the groundwork for investigating the use of fuzzy set theory in this process. Since ontology matching is a form of automated reasoning, many of the techniques in approximate reasoning could be useful. The generic matching capabilities of some systems such as GLUE could be used and extended to examine how different fuzzy similarity measures and aggregation operators [6] substituted at the element and structure-levels might affect the quality of the ontology matching process.

An alternative to the ontology matching process is the specification of articulation rules that describe how similar concepts are related or translated between different ontologies. One of the problems is as the ontologies grow and change, the articulation rules need to be updated. The use of approximate reasoning methods in automating the maintenance of articulation rules between ontologies might be investigated.

6. References

[1] Bergamaschi, S. Castano, S. De Capitani di Vermercati, S. Montanari, and M. Vicini, “An Intelligent Approach to Information Integration,” Proc. First Int’l Conf. Formal Ontology in Information Systems, N. Guarino, ed., pp. 253-268, 1998.

[2] Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. Scientific American, 279, 2001.

[3] M. Bright, A. Hurson, and S. Pakzad, “Automated Resolution of Semantic Heterogeneity in Multidatabases,” ACM Trans. Database Systems, vol. 19, pp. 212-253, 1994

[4] Chaudhri, V.K., Farquhar, A., Fikes, R., Karp, P.D. and Rice, J.P. (1998). OKBC: A Programmatic Foundation for Knowledge Base Interoperability. In: *Proceedings of theFifteenth National Conference on Artificial Intelligence (AAAI-98*), Madison, Wisconsin, AAAI Press.

[5] C. Collet, M. Huhns, and W. Shen, “Resource Integration Using a Large Knowledge Base in Carnot,” Computer, vol. 24, pp. 55-62,1991.

[6] V. Cross and T. Sudkamp, Similarity and Compatibility in Fuzzy Set Theory: Assessment and Applications, New York: Physica-Verlag, ISBN 3-7908-1458, 2002

[7] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, “Learning to map between ontologies on the semantic web”, In *Proceedings of the Eleventh International Conference on World Wide Web*, pages 662–673. ACM Press, 2002.

[8] Frakes and R. Baeza-Yates, eds., *Information Retrieval: Data Structures and Algorithm*s, Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[9] A. Gal, G. Modica, and H.M. Jamil. “Improving web search with automatic ontology matching” Submitted for publication. Available citeseer.nj.nec.com/gal03improving.html, 2003.

[10] A. Gangemi, D. Pisanelli, and G. Steve, “Ontology Integration: Experiences with Medical Terminologies,” Formal Ontology in Information Systems, N. Guarino, ed., pp. 163-178, 1998.

[11] Gruber, T. R.. “A Translation Approach to portable ontology Specifications”, *Knowledge Acquisition*, 5(2) 1993, pp. 199-220.

[12] J. Lee, M. Kim, and Y. Lee, “Information Retrieval Based on Conceptual Distance in IS-A Hierarchies,” J. Documentation, vol. 49,pp. 188-207, 1993.

[13] D. Lenat and R. Guha, Building Large Knowledge Based Systems: Representation and Inference in the CYC Project. Reading, Mass.:Addison-Wesley Publishing Company, 1990.

[14] Levenshtein, V. "Binary Codes capable of correcting deletions, insertions, and reversals," *Cybernetics and ontrol Theor*y, 10(8):707 -710, 1966.

[15] J. Madhavan, P. A. Bernstein, and E. Rahm, "Generic Schema Matching with Cupid,"*Proceedings of 27th International Conference on Very Large Data Bases (VLDB*),P. M. G. Apers, P. Atzeni, S. Ceri, S. Paraboschi, K. Ramamohanarao, and R. T.Snodgrass, eds., Roma, Italy, September 2001, pp. 49 - 58, Morgan Kaufmann.

[16] Alexander Maedche and Steffen Staab,"Comparing Ontologies - Similarity Measures and a Comparison Study" Institute AIFB, University of Karlsruhe, Internal Report, Available citeseer.nj.nec.com/article/maedche01comparing.html, 2001.

[17] Melnik, S., H. Garcia-Molina, E. Rahm: Similarity Flooding: A Versatile Graph Matching Algorithm. *ICDE*, 2002.

[18] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller,"Introduction to WordNet: An On-Line Lexical Database," Int'l J.Lexicography, vol. 3, pp. 235-244, 1990.

[19] R.J. Miller, L.M. Haas, and M.A. Hern´ andez. "Schema mapping as query discovery". In A. El Abbadi, M.L. Brodie, S. Chakravarthy, U. Dayal, N. Kamel, G. Schlageter, and K.-Y. Whang, editors, *Proceedings of the International conference on very Large Data Bases (VLDB*), pages 77–88. Morgan Kaufmann, 2000.

[20] N. Noy and M. Musen. *PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment* In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000*), pp. 450-455, 2000.

[21] N. F. Noy and M. A. Musen "Anchor-PROMPT: Using non-local context for semantic matching". In *Proc. of the IJCAI-2001 Workshop on Ontologies and Information Sharin*g, Seattle, WA,August 2001.

[22] Erhard Rahm 1 ,Philip A.Bernstein, "A survey of approaches to automatic schema matching." The VLDB Journal vol. 10, 2001, pp. 334 350.

[23] O. Resnik, "Semantic Similarity in a Taxonomy: An Information-Based Measure and Its Application to Problems of Ambiguity and Natural Language," J. Artificial Intelligence Research, vol. 11, pp. 95-130, 1999.

[24] M. Rodriguez, M. Egenhofer, "Determining Semantic Similarity among Entity Classes from different ontologies," IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 2, pp. 442-456, 2003.

[26] A. Sheth, "Changing Focus on Interoperability in InformationSystems: From System, Syntax, Structure to Semantics," Interoperating Geographic Information Systems, M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, eds., pp. 5-30, 1999.

[27] Steffen Staab et al, "Ontologies' KISSES in Standardization", IEEE Intelligent Systems, March/April, pp. 70-79, 2002.

[28] Tversky, "Features of Similarity," Psychological Rev., vol. 84, pp. 327-352, 1977.

[29] Uschold, M.; Gringer, M. "ONTOLOGIES: Principles, Methods, and Applications*" Knowledge Engineering Review*. Vol. 11. N. 2. June. 1996.

[30] M. Uschold. "Where is the semantics in the Semantic Web?", In Workshop on Ontologies in Agent Systems (OAS) at the 5th International Conference on Autonomous Agents, 2001.

[31] H. Wache, T. Voegele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Huebner. "Ontology-based integration of information - a survey of existing approaches". In *Proc. of IJCAI*, August 2001.

[31] P. Weinstein and P. Birmingham, "Comparing Concepts in Differentiated Ontologies," Proc. 12th Workshop Knowledge Acquisition, Modeling, and Management, 1999.