Semantic Integration: A Survey Of Ontology-Based Approaches

Natalya F. Noy
Stanford Medical Informatics
Stanford University
251 Campus Drive, Stanford, CA 94305
noy@smi.stanford.edu

ABSTRACT

Semantic integration is an active area of research in several disciplines, such as databases, information-integration, and ontologies. This paper provides a brief survey of the approaches to semantic integration developed by researchers in the ontology community. We focus on the approaches that differentiate the ontology research from other related areas. The goal of the paper is to provide a reader who may not be very familiar with ontology research with introduction to major themes in this research and with pointers to different research projects. We discuss techniques for finding correspondences between ontologies, declarative ways of representing these correspondences, and use of these correspondences in various semantic-integration tasks

1. ONTOLOGIES AND SEMANTIC INTEGRATION

Researchers and practitioners in the fields of databases and information integration have produced a large body of research to facilitate interoperability between different systems. This research ranges from techniques for matching database schemas to answering queries using multiple sources of data. Ontology research is another discipline that deals with semantic heterogeneity in structured data. We refer the reader to another article in this issue [24] for a detailed discussion on the uses of ontologies, their differences from database schemas, and challenges in semantic integration that the ontology community faces. The goal of this paper is to discuss the major thrusts of approaches to semantic integration produced by various projects in the ontology community and to provide readers with pointers to sources for additional information. We will focus on the approaches that highlight the use of ontologies, their emphasis on knowledge sharing, and their use in reasoning. Note that this paper does not attempt to provide a comprehensive review of the state of the art in using ontologies for semantic integration. We refer the reader to an excellent and thorough review by Kalfoglou and Schorlemmer [15] for that purpose.

While there are many definitions of what an ontology is [26], the common thread in these definitions is that an ontology is some formal description of a domain of discourse, intended for sharing among different applications, and expressed in a language that can be used for reasoning. These features of ontologies underscore the main trends that distinguish semantic-integration research in the ontology community: First, since the underlying goal of ontology development is to create artifacts that different applications can share, there is an emphasis on creating common ontologies that can then be extended for more specific domains and applications. If these extensions refer to the same top-level ontology, the problem of integrating them can be greatly alleviated. Second, since ontologies are developed for use with reasoning engines and semantics of ontology languages are specified with reasoning in mind, inference and reasoning takes center stage in ontology-integration approaches.

Ontologies have gained popularity in the AI community as a means for establishing explicit formal vocabulary to share between applications. Therefore, one can say that one of the goals of using ontologies is not to have the problem of heterogeneity at all. It is of course unrealistic to hope that there will be an agreement on one or even a small set of ontologies. While having some common ground either within an application area or for some high-level general concepts could alleviate the problem of semantic heterogeneity, we will still need to map between ontologies, whether they extend the same top-level ontology or are developed independently.

So, what are the types of differences between ontologies? In part summarizing earlier surveys, Klein [16] categorizes different types of mismatches between ontologies. The first class of mismatches are mismatches at the *language level*—mismatches in expressiveness and semantics of ontology language. The languages can differ in their syntax, but, more important, constructs available in one language (e.g., stating that classes are disjoint) are not available in another. Even semantics of the same language primitives could be different (e.g., whether declaration of multiple ranges of a property have union or intersection semantics). The *normalization* process therefore often precedes ontology-matching [15] and translates source ontologies to the same language, resolving these differences.

However, even for ontologies expressed in the same language, possible *ontology-level* mismatches abound. A partial list of ontology-level mismatches includes using the same linguistic terms to describe different concepts; using different terms to describe the same concept; using different modeling paradigms (e.g., using interval logic or points for temporal representation); using different modeling conventions and levels of granularity; having ontologies with differing coverage of the domain, and so on.

We discuss three dimensions of semantic-integration research in this paper:

Mapping discovery: Given two ontologies, how do we find similarities between them, determine which concepts and properties represent similar notions, and so on.

Declarative formal representations of mappings: Given two ontologies, how do we represent the mappings between them to enable reasoning with mappings.

Reasoning with mappings: Once the mappings are defined, what do we do with them, what types of reasoning are involved?

In the rest of this paper, we explore these dimensions.

2. DISCOVERING MAPPINGS

Many researchers agree that one of the major bottleneck in semantic integration is mapping discovery. There are simply too many ontologies and database schemas available and they are too large to have manual definition of correspondences as the primary source of mapping discovery. Furthermore, in the world where software agents will roam the (semantic) web, they will need to map structures they know about to new structures they come across on-the-fly. Hence, the task of finding mappings (semi-) automatically has been an active area of research in both database and ontology communities [22, 15].

We identify two major architectures for mapping discovery between ontologies. For the first approach, recall that the goal of ontologies is to facilitate knowledge sharing. As a result, ontologies are often developed with the explicit goal of providing the basis for future semantic integration. Here, the vision is that a general upper ontology is agreed upon by developers of different applications, who then extend this general ontology with concepts and properties specific to their applications. As long as this extension is performed in a way consistent with the definitions in the shared ontology, finding correspondences between two extensions can be facilitated by this common "grounding." The second set of approaches comprises heuristics-based or machine learning techniques that use various characteristics of ontologies, such as their structure, definitions of concepts, instances of classes, to find mappings. These approaches are similar to approaches to mapping XML schemas or other structured data (e.g., Cupid [17]) but tend to rely more heavily on features of concept definitions or on explicit semantics of these definitions.

2.1 Using a Shared Ontology

A number of very general ontologies formalizing notions such as processes and events, time and space, physical objects, and so on, are being developed and some of them are becoming accepted standards. The explicit goal of these ontologies is to have domain-specific ontologies extend them, thus providing the grounding in common vocabulary for these ontologies. Note that this scenario is different from the traditional information-integration scenario where the global schema—the common view on different schemas to be integrated—is usually developed *after* the schemas themselves are developed and its design is therefore guided by the individual schemas to be integrated. The implication of this difference is that in the information-integration scenario the global schema is only general enough to provide access to all the schemas that it integrates. The common top-level or reference ontology is usually more general since it needs to encompass the top level for ontologies yet to be developed.

Two of the ontologies that are built specifically with the purpose of being formal top-level ontologies are the Suggested Upper Merged Ontology (SUMO) [19] and DOLCE [8]. SUMO is an effort by the IEEE Standard Upper Ontology Working Group aimed at developing "a standard upper ontology that will promote data interoperability, information search and retrieval, automated inferencing, and natural language processing." The SUMO ontology defines such high-level concepts Object, ContinousObject, Process, Quantity, Relation, and so on, providing axioms in first-order logic that describe properties of these concepts and relations among them. Similarly, the DOLCE ontology is a formal foundational ontology developed as a top-level ontology in the WonderWeb project, which comprises a large number of European research groups. The goal of DOLCE is to provide a common reference framework for WonderWeb ontologies to facilitate sharing of information among them. In its representation, DOLCE aims at capturing "ontological categories underlying natural language and human common-sense."

While many researchers hope that domain- and application-specific ontologies will reuse the foundational ontologies, like SUMO and DOLCE, and that such reuse will indeed facilitate semantic interoperation between applications based on these ontologies, we do not yet have enough experience reports with such approaches to claim it a success. There are reports on both the successes [21] and difficulties [25] of such reuse. The Workshop on Core Ontologies in Ontology Engineering[1] in October 2004 will discuss both successful and unsuccessful cases and best practices on reusing foundational ontologies for specifying domain content.

There are also implemented semantic-integration tools that exploit the idea that if two ontologies extend the same reference ontology in a consistent way, then finding correspondences between their concepts is easier. For example, the Process Specification Language (PSL) [11], developed at the National Institute for Standards and Technology, is an ontology that is endorsed as an International Standard within the International Organization of Standardisation (ISO). PSL was designed to "facilitate correct and complete exchange of process information among manufacturing systems such as

[1] www.loa-cnr.it/core_onto.html

scheduling, process modeling, [and] process planning" [12]. The designers of PSL have developed it as an interlingua for ontologies representing these different process. All theories within the PSL ontology have been verified with respect to the intended semantics of their terminology. Grüninger and Kopena [12] developed an integration architecture with the PSL ontology at the center and mappings between ontologies for specific manufacturing processes and the PSL ontology. The mappings are defined semi-automatically by presenting ontology developers with a set of questions (in natural language) helping them to map terms in their process-specific ontology to the terms in PSL. The system then generates two-way mappings between the task-specific ontology, such as scheduling and the PSL interlingua. Note that the generation of these mappings is defined formally and is not based on heuristics. These mappings can be composed to provide mappings between any task-specific ontologies.

2.2 Using Heuristics and Machine-learning

It is certainly helpful to have ontologies that we need to match to refer to the same upper ontology or to conform to the same reference ontology. However, we often do not have this "luxury" and need to create mappings between ontologies that perhaps use the same specification language but do not have any vocabulary beyond the specification language in common. Most researchers agree that automatic mapping between ontologies in this context is beyond our grasp at the moment, but many techniques have produced good results.

Heuristic-based approaches to ontology mapping are similar to heuristic-based approach to matching database schemas and XML structures ([22, 17]) and use lexical and structural components of definitions to find correspondences. However, ontology-based approaches often go further, exploiting semantics of relationships in ontologies, such as, for example, the semantics of the subclass-of or part-of relationships, attachment of property to a class, domain and range definitions for properties, and so on. Ontologies usually have a lot more constraints specified than database schemas do, and the methods for finding mappings automatically tend to exploit this larger number of constraints

We will start by reviewing several ontology-mapping tools and then summarize the different ontology features that they use. We would like to emphasize again that this paper presents only a sampling of such tools to give examples of different approaches. Please see a paper by Kalfoglou and Schorlemmer [15] for a comprehensive review.

Hovy [13] describes a set of heuristics that researchers at ISI/USC used for semi-automatic alignment of domain ontologies to a large central ontology. Their techniques are based mainly on linguistic analysis of concept names and natural-language definitions of concepts. (There is a limited use of taxonomic relationships as well). First, the matcher uses natural-language–processing techniques to split composite-word names (a common occurrence in concept names). It then compares substrings of different lengths to find concept names that are similar to each other. The second consideration are the words used in natural-language definitions of concepts. The matcher compares the number and the ratio of shared words in the definitions to find definitions that are similar. An experimentally determined formula for combining these measures of similarity yields potential matchers that the user needs to examine and approve.

The PROMPT system [20] was originally developed to support ontology merging, guiding users through the process and suggesting which classes and properties can be merged. It records the mappings identified both by the system and by the user during merging to create a declarative mapping specification between source ontologies. To make suggestions, PROMPT uses a mixture of lexical and structural features, as well as input from the user during an interactive merging session to find the mappings. For instance, if a user said that two classes in two source ontologies are the same (should be merged), then PROMPT analyzed the properties of these classes, their subclasses and superclasses to look for similarities of their definitions and suggest additional correspondences. Another algorithm in the toolset–ANCHORPROMPT [20]—treats an ontology as a graph with classes as nodes and slots as links. The algorithm analyzes the paths in the subgraph limited by the anchors and determines which classes frequently appear in similar positions on similar paths. These classes are likely to represent semantically similar concepts.

Recently, the W3C has approved a standard for representing ontologies on the Semantic Web—the OWL language.[2] Acceptance of a standard encouraged researchers to propose algorithms that rely more heavily on features of the ontology language to compare ontologies. For example, a similarity metric between concepts in OWL ontologies developed by Euzenat and Volchev [7] is a weighted combination of similarities of various features in OWL concept definitions: their labels, domains and ranges of properties, restrictions on properties (such as cardinality restrictions), types of concepts, subclasses and superclasses, and so on.

FCA-Merge [23] is a method for comparing ontologies that have a set of shared instances or a shared set of documents annotated with concepts from source ontologies. Based on this information, FCA-Merge uses techniques from Formal Concept Analysis [9] to produce a lattice of concepts which relates concepts from the source ontologies. The algorithm suggests equivalence and subclass–superclass relations. An ontology engineer can then analyze the result and use it as a guidance for creating a merged ontology.

The IF-Map [14] system identifies mappings automatically based on the theory of information flow [1]. Given two ontologies, IF-Map generates a *logic infomorphism*—a mapping between ontologies that is based on the above conformance. The system then uses the channel theory to infer the mappings between different local ontologies using these logic infomorphisms.

[2] `http://www.w3.org/TR/owl-features/`

GLUE [5] is an example of a system that employs machine-learning techniques to find mappings. GLUE uses multiple learners exploiting information in concept instances and taxonomic structure of ontologies. GLUE uses a probabilistic model to combine results of different learners. The learners that GLUE uses currently relies on ontologies having instances and they work much better if many slot values have text in them rather than references to other instances.

Researchers have also addressed the issue of finding complex mappings, such as determining that a concepts in one ontology is a specialization of a concept in another ontology. For example, Giunchiglia and Shvaiko [10] start by grounding their source ontologies in WordNet terms but then run a SAT prover on the mappings to determine other types of mappings (such as generalization, specialization or disjointness): the authors reformulate the matching problem as that of propositional satisfiability.

To summarize, the tools for automatic and semi-automatic ontology alignment use the following features in ontology definitions (to various extent):

- concept names and natural-language descriptions
- class hierarchy (subclass–superclass relationships)
- property definitions (domains, ranges, restrictions)
- instances of classes
- class descriptions (as in DL-based tools).

3. REPRESENTATIONS OF MAPPINGS

While developing tools for automatic and semi-automatic ontology matching is a large thrust of semantic-integration research in the ontology community, it is definitely not the only one. The higher expressive power of ontology languages provides the opportunity for representing mappings themselves in more expressive terms. Mappings between elements in schemas are usually expressed either as queries and views or as pairs of related terms. We generally find a larger spectrum of the ways mapping between ontologies are expressed. We will discuss several representations of mappings here: representing mappings as instances in an ontology of mappings; defining bridging axioms in first-order logic to represent transformations; and using views to describe mappings from a global ontology to local ontologies.

In the OntoMerge system [6] developed for semantic integration on the Semantic Web, authors use a general-purpose inference engine to enable translation between mapped ontologies. In OntoMerge the correspondence between two ontologies is expressed as a set of *bridging axioms* relating classes and properties of the two source ontologies. The vocabulary of the two ontologies are in different XML namespaces, so the bridging axioms are essentially translation rules referring to concepts from source ontologies and specifying how to express for example a class in one ontology by collecting information from classes in another. The two source ontologies, together with the bridging axioms are then treated as a single theory by a theorem prover optimized for ontology-translation task. The theorem prover runs either in forward-chaining or backward-chaining mode depending on the task at hand.

Several researchers use ontologies themselves to represent mappings declaratively, as instances in an ontology. The *mapping ontology* by Crubézy and colleagues [4] or the *Semantic Bridge Ontology* of the MAFRA framework [18], for instance, define the structure of specific mappings and the transformation functions to transfer instances from one ontology to another. This ontology can then be used by tools to perform the transformations. Such an ontology usually provides different ways of linking concepts from the source ontology to the target ontology, transformation rules to specify how values should be changed, and conditions and effects of such rules. Then a mapping between two ontologies constitutes a set of instances of classes in the mapping ontology and can be used by applications to translate data from the source ontology to the target. The mapping ontology mentioned above [4], for example, provides declarative means for defining many-to-one or many-to-many aggregation relationships between concepts in the source and target ontologies, as well as one-to-many concept-decomposition relations. It allows specification of recursive mappings, complex mappings between that collect information from several related concepts, and other mechanisms.

Finally, researchers also used views to define mappings between ontologies, similar to defining mappings in information integration, both in global-as-view (GAV) and local-as-view (LAV) setting. The OIS framework [2] is a good example of such approach. In OIS, a global ontology is used to provide access to local ontologies. Both global and local ontologies are defined using Description Logics. The mappings are defined as views over either the global or the local ontologies. In other words, a predicate from one ontology is defined as a query (and DL expression) over predicates in another ontology.

4. WE HAVE THE MAPPINGS. NOW WHAT?

Naturally, defining the mappings between ontologies, either automatically, semi-automatically, or interactively, is not a goal in itself. The resulting mappings are used for various integration tasks: data transformation, query answering, or web-service composition, to name a few.

Given that ontologies are often used for reasoning, it is only natural that many of these integration tasks involve reasoning over the source ontologies and the mappings. For example, the OntoMerge system mentioned earlier [6] uses reasoning to perform several tasks related to ontology translation. The first task is translating instances that conform to one ontology (the source) to instances conforming to another ontology (the target), given the mapping between the source and target. To perform this task, OntoMerge first creates a merged ontology that includes the source, the target, and the mapping and performs inference on this merged ontology.

Afterwards, OntoMerge performs a projection step, where it retains only the new conclusions reached that exclusively reference the target vocabulary.

The second task that OntoMerge deals with—generating ontology extensions—is more specific to the area of ontologies. Consider for example, two ontologies describing Web services: OWL-S[3] and WSDL[4]. Suppose we have defined a mapping between these two ontologies. Suppose also that we have an ontology describing ticket-purchasing web services—a domain-specific extension of the OWL-S ontology. This ticket-purchasing ontology creates subclasses of some of the classes in OWL-S, fills in some of the property values, and so on. In other words, it *extends* the OWL-S ontology. If we have a mapping between OWL-S and WSDL, OntoMerge can automatically generate a WSDL description of ticket-purchasing—an extension of the WSDL ontology. Note that this case is different from data translation since we are dealing with subontologies rather than instances conforming to ontologies. In both of these tasks, OntoMerge uses forward-chaining reasoner to perform the translation.

Several tools process representation of mappings as instances of the mapping ontology by Crubézy and colleagues discussed in the previous section [4] to perform various integration tasks. First, a *mapping interpreter* uses the instances in the mapping ontology to translate data from the source ontology to the target ontology. Second, the PROMPT tool for ontology merging [20], also mentioned earlier in the paper, can take the mapping ontology as its input and merge the ontologies based on the mapping. These tools are extensions to the Protégé ontology-development environment[5] thus providing an integrated framework for ontology development, knowledge acquisition, and semantic integration.

In the OIS framework [2], ontologies are expressed in Description Logics and therefore it is natural that DL reasoners are used to answer queries in the data-integration framework. The authors address the general task of answering queries posed in terms of the global ontology using the data in the local ontologies. However, while even in expressive Description Logics, computing certain answers to queries is decidable, it may often be intractable. In recent research, the authors have explored less expressive subsets of Description Logics [3], making this type of query answering tractable.

5. CONCLUDING REMARKS

As this brief survey shows, many issues that ontology researchers in semantic integration grapple with are very similar to the issues that database and information-integration researchers have been addressing. Some of the approaches are also similar although the ontology community relies more heavily on the higher expressive power of ontology languages and on reasoning techniques. With ontologies, using a common upper ontology or reference ontology to alleviate the integration problem is also a common approach.

The two communities can certainly share and reuse the techniques that they have developed in their respective domains. In fact, there has been a certain convergence trend where schema-matching approaches for example employ more expressive components of schema definitions in their techniques. On the other hand, ontology researchers are paying more attention to the experience of the database community. We believe that such cross-fertilization will improve semantic-integration solutions in both fields.

Finally, the emerging Semantic Web can prove to be an excellent testbed for scalability of various approaches and a common ground for experimenting with hybrid approaches. Most researchers agree that semantic integration is one of the most serious challenges for the Semantic Web today. On the one hand, the premise of the Semantic Web is that the use of machine-interpretable ontologies defined in formal languages amenable to reasoning will provide the next generation of services. On the other hand, the scale of the Semantic Web will certainly require well-tested approaches from the database community.

6. ACKNOWLEDGMENTS

I would like to thank AnHai Doan, Michael Grüninger, and Yannis Kalfoglou for thoughtful and helpful comments on earlier drafts at this paper.

REFERENCES

[1] J. Barwise and J. Seligman. *Information Flow: The Logic of Distributed Systems*. Cambridge University Press, 1997.

[2] D. Calvanese, G. Giacomo, and M. Lenzerini. Ontology of integration and integration of ontologies. In *Description Logic Workshop (DL 2001)*, pages 10–19, 2001.

[3] D. Calvanese, G. D. Giacomo, M. Lenzerini, R. Rosati, and G. Vetere. DL-lite: Practical Reasoning for Rich DLs. In *Description Logic Workshop (DL2004)*, Whistler, Canada, 2004.

[4] M. Crubézy and M. A. Musen. Ontologies in support of problem solving. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, pages 321–342. Sringer, 2003.

[5] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In *The Eleventh International WWW Conference*, Hawaii, US, 2002.

[6] D. Dou, D. McDermott, and P. Qi. Ontology translation on the semantic web. In *International Conference on Ontologies, Databases and Applications of Semantics*, 2003.

[7] J. Euzenat and P. Valtchev. Similarity-based ontology alignment in OWL-Lite. In *The 16th European Conference on Artificial Intelligence (ECAI-04)*,

[3] `http://www.daml.org/services/owl-s/1.0/`

[4] `http://www.w3.org/TR/wsdl`

[5] `http://protege.stanford.edu`

Valencia, Spain, 2004.

[8] A. Gangemi, N. Guarino, C. Masolo, and A. Oltramari. Sweetening wordnet with DOLCE. *AI Magazine*, 24(3):13–24, 2003.

[9] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical foundations*. Springer, Berlin-Heidelberg, 1999.

[10] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Semantic matching. In *1st European semantic web symposium (ESWS'04)*, Greece, 2004.

[11] M. Grüninger. A guide to the ontology of the process specification language. In S. Staab and R. Studer, editors, *Handbook on Ontologies*. Sringer, 2003.

[12] M. Grüninger and J. Kopena. Semantic integration through invariants. In *Workshop on Semantic Integration at ISWC-2003*, Sanibel Island, FL, 2003.

[13] E. Hovy. Combining and standardizing largescale, practical ontologies for machine translation and other uses. In *The First International Conference on Language Resources and Evaluation (LREC)*, pages 535–542, Granada, Spain, 1998.

[14] Y. Kalfoglou and M. Schorlemmer. IF-Map: an ontology mapping method based on information flow theory. *Journal on Data Semantics*, 1(1):98–127, Oct. 2003.

[15] Y. Kalfoglou and M. Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1–31, 2003.

[16] M. Klein and D. Fensel. Ontology versioning on the semantic web. In *The First Semantic Web Working Symposium*, Stanford, CA, 2001.

[17] J. Madhavan, P. Bernstein, and E. Rahm. Generic schema matching using Cupid. In *The 27th International Conf. on Very Large Data Bases (VLDB '01)*, Rome, Italy, 2001.

[18] A. Maedche, B. Motik, N. Silva, and R. Volz. MAFRA - a mapping framework for distributed ontologies. In *13th European Conference on Knowledge Engineering and Knowledge Management EKAW*, Madrid, Spain, 2002.

[19] I. Niles and A. Pease. Towards a standard upper ontology. In *The 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, Ogunquit, Maine, 2001.

[20] N. F. Noy and M. A. Musen. The PROMPT suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.

[21] S. Polyak, J. Lee, M. Gruninger, and C. Menzel. Applying the process interchange format(PIF) to a supply chain process interoperability scenario. In *Workshop on Applications of Ontologies and Problem Solving Methods, ECAI'98*, Brighton, England, 1998.

[22] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4), 2001.

[23] G. Stumme and A. Mädche. FCA-Merge: Bottom-up merging of ontologies. In *7th Intl. Conf. on Artificial Intelligence (IJCAI '01)*, pages 225–230, Seattle, WA, 2001.

[24] M. Uschold and M. Grüninger. Ontologies and semantics for seamless connectivity. *SIGMOD Record*, 33(3), 2004.

[25] A. Valente, T. Russ, R. MacGrecor, and W. Swartout. Building and (re)using an ontology for air campaign planning. *IEEE Intelligent Systems*, 14(1):27–36, 1999.

[26] C. Welty. Ontology research. *AI Magazine*, 24(3), 2003.