Ontology Negotiation between Scientific Archives

Sidney C. Bailin
Knowledge Evolution, Inc.
sbailin@kevol.com

Walt Truszkowski
NASA/Goddard Space Flight Center
Walt.Truszkowski@gsfc.nasa.gov

Abstract

This paper describes an approach to ontology negotiation between information agents. Ontologies are declarative (data driven) expressions of an agent's "world": the objects, operations, facts, and rules that constitute the logical space within which an agent performs. Ontology negotiation enables agents to cooperate in performing a task, even if they are based on different ontologies. The process allows agents to discover ontology conflicts and then, though incremental interpretation, clarification, and explanation, establish a common basis for communicating with each other.

1. Introduction

The past several years have witnessed a proliferation of information sources on the world-wide-web, and of information agents with widely varying specialties. The unmanageability of massive amounts of information is becoming apparent and is having an impact on professions that rely on distributed archived information.

Ontology negotiation is becoming increasingly recognized as a crucial element of scalable agent technology. This is because agents, by their very nature, are supposed to operate with a fair amount of autonomy and independence from their end-users. Part of this independence is the ability to enlist other agents for help in performing a task (such as locating information on the web). The agents enlisted for help may be "owned" by a different end-user or organization (such as a document archive), and there is no guarantee that they will use the same terminology or understand the same concepts (objects, operators, theorems, rules) as the recruiting agent.

For NASA, the need for ontology negotiation arises at the boundaries between scientific disciplines. For example: modeling the effects of global warming might involve knowledge about imaging, climate analysis, ecology, demographics, industrial economics, and biology. The need for ontology negotiation also arises at the boundaries between scientific programs. For example, a Principal Investigator may want to use information from a previous mission to complement downloads from the instruments currently deployed.

1.1 Summary of achievements

We have developed an ontology negotiation protocol (ONP) and a framework for implementing agents that use the protocol. We have created a testbed in which a user agent and agents representing two large earth science archives cooperate to improve information retrieval performance. Specifically, the testbed can handle queries that match documents in either or both archives but are not formulated in terms of either archive's taxonomy. The translation knowledge involved in satisfying the query is then added to the archives' ontologies, thereby improving future performance. For these experiments we have used NASA's Global Change Master Directory and NOAA's Wind and Sea Index.

The absence of an explicit ontology in these (and most existing) archives presents an obvious challenge to the work. We have addressed it in the short term by deriving lightweight ontologies from the classification pages provided by each archive's web interface. In order to support the ONP, an ontology must provide more than a topic classification; it must provide answers to certain questions concerning synonyms and relations between topics. We have articulated these questions in the form of an application program interface (API) that an ontology must implement in order to support the ONP. For the purpose of experimentation, we developed simple forms of these functions for both the NASA and the NOAA archives.

In the longer term, we expect that web-based knowledge management techniques (involving XML, RDF, topic maps, or similar ideas) will gain widespread currency. This will enable automated agents to obtain ontology information from scientific archives through something resembling our current API.

2. Ontology conflict and negotiation

The system employs the WordNet lexical database [4] as a source of extending each ontology's concept repertoire. However, the heart of the process lies in the exchanges between agents when WordNet by itself does not allow the agents to interpret each other's concepts. The exchanges, structured by the rules of the ONP, allow each agent to ask for clarification of previous messages and for confirmation or correction of attempted interpretations. Interpretation and clarification may take the form of simple substitution of synonyms, but the protocol provides for more complex forms such as formal logical definitions, operational descriptions (i.e., rules governing the use of concepts), and approximations to a concept's meaning.

At the same time, we have drawn a boundary between these exchanges and the way in which an agent modifies its ontology. The boundary takes the form of an application program interface (API) that specifies several ways of querying and evolving an ontology, but leaves the implementation of the operations unspecified. The semantics of the evolution operations are specified only insofar as they are reversed directions of the clarification and interpretation methods. For example, the operation `addSlightGeneralization` reverses the `getSlightGeneralization`. The semantics of the interpretation and clarification methods, also part of the ontology API, are to some extent specified by the rules of the ONP. There are logical barriers to completely specifying the semantics of these operations since that would be tantamount to defining truth, which cannot be done formally [12].

The notion of defining an API to an ontology deserves some explanation. Usually an ontology is considered as a relatively passive entity. Ontology authoring frameworks provide inference and computations to support the construction of an ontology and to perform queries against it, but the ontology itself is usually distinguished from this set of functions. In our work we are exploring the possibility of ongoing evolution of an ontology under the control of an autonomous agent. In the long-term vision, agents rather than humans are the primary authors and users of an evolving ontology, even if they start from a human-authored one. To support this, the ontology must reside within a framework that allows agents not only to query but also to modify the ontology. The API we have defined may be viewed as a minimal set of requirements for such a framework. Our belief that it is minimal is based on the kinds of functionality typically provided to the user by today's ontology authoring systems.

Our contribution is therefore not in a novel way of resolving ontology conflicts, which is an issue being addressed by work in ontology construction [1,2,10,11], semantics for world-wide-web searches [3,7.8], and mediation in information retrieval [13]. The ONP is, rather, a set of rules for cooperatively attacking the problem when it arises during agent communication. Consider what happens when two people try to negotiate an ontology conflict. They try to elicit from each other information that allows each of them to understand the other's utterances. It is ultimately up to each person to decide what to do with the new information, e.g., to assimilate a new concept, to place it in relation to previously known concepts, to create a distinction not previously acknowledged, etc. The ONP is an attempt to automate the inter-agent dialogue. It neither replaces the necessary methods for modifying an ontology, nor is it subsumed by those methods.

As a protocol for agent communication, the ONP is related to standards such as the Knowledge Query Markup Language [5] and the Federation for Intelligent Physical Agents (FIPA) Agent Communication Language [6]. The ONP is built on top of KQML, and could easily be adapted to FIPA ACL. In addition, while this is not currently the case, the Knowledge Interchange Format [9] could be used to represent the logical descriptions that are exchanged within these messages.

3. Overview of the Protocol

Our goal was to specify and implement a robust method of ontology negotiation that allows web-based information agents to resolve semantic mismatches in real time without human intervention. The resulting software provides "strange" agents with a means of arriving at a common language in which to converse. "Common language" refers not just to syntax, but also to the *meaning* of terms exchanged by the agents. The meaning of a term can be represented by an ontology in any of several forms: e.g., as facts pertaining to the term, as rules governing the use of the term, or as structural information relating the term to other terms (e.g., the "is-a" relation).

Between information agents, the "terms exchanged" consist primarily of the *query content* and *document descriptors* for the query results. Both of these can be viewed as keywords describing the document that is either desired (query) or found (results). Our research to date indicates that there are three processes involved in this type of negotiation:

- *Clarifying* the meaning of keywords
- *Explaining* the relevance of the query results
- *Evolving* an agent's ontology on the basis of clarification and relevance explanation

In addition, agents need to be able to locate other agents capable of performing specific search, fusion, or filtering tasks. Other researchers have extensively explored the mechanisms for discovering an agent with specific capabilities and engaging it in a task. Their research, however, has tended to ignore the problems of semantic mismatch: When an agent announces that it has (or needs) a particular capability, will other agents necessarily understand the terms in which the capability is described? The question suggests that semantic clarification, and therefore ontology negotiation, should play a role in the negotiation of capabilities and assignments, as much as in the satisfaction of information requests.

3.1 Objects of the protocol

The ontology negotiation protocol (ONP) is based on a small set of object types and operations on those objects. The available object types are presented in Table 1; some of them warrant explanation. *Queries* are requests to locate documents and/or URLs that are relevant to a set of descriptors (keywords). *Declination* expresses an agent's unwillingness to respond to a query (for reasons of capacity or capability), while *rejection* is an agent's expression of dissatisfaction with the current results of a query.

Confirmation of interpretation is an agent A's validation of an agent B's tentative understanding of a previous message from A. *Clarification* is the means by which an agent makes explicit the meaning of a previous message it sent.

An agent receives the results of a query as a set of URLs and document descriptors. It uses the descriptors (keywords) to *evaluate* the relevance of the results to the query. If it cannot see why a particular URL is relevant to the query, it can ask the agent that returned the results for an *explanation of relevance*. The structure of such an explanation is discussed in Section 3.3; here we just note that the explanation may include facts that are true in the server agent's ontology. When the querying agent receives an explanation of relevance, but it cannot derive a particular fact used in the explanation, it can request the server agent for an *explanation of fact*.

3.2 States and transitions of the protocol

States of the straw-man protocol correspond to performance of one or another operation on a particular type of object. The available operations are shown in Table 2.

Obviously not every combination of an operation and object represents a meaningful state. Table 3 presents the states of the ONP, using the abbreviations listed in Tables 1 and 2. For example, an agent can *wait* for a *capability statement* (if it has requested one) but it cannot *wait* for a *declination*. Similarly, an agent can *receive* a *declination* from another agent, but it cannot *receive* an *ontology*.

The behavior of agents participating in the protocol is determined by transitions between the possible states. Since there are 46 states defined in the straw-man protocol, the transitions are selected from the 46 x 46 element matrix of state pairs.

The range of possible transitions presents a challenge in implementing the protocol—especially to implement it in a maintainable fashion. We have addressed this problem through a tool that provides a high-level table-based means of specifying the states and transitions.

Object Type	**Abbreviation**	**Object Type**	**Abbreviation**
Query	Que	Confirmation of Interpretation	Cfi
Query Results	Qre	Clarification	Cla
Acknowledgement	Ack	Explanation of Relevance	Exr
Rejection	Rej	Explanation of Fact	Exf
Declination	Dcl	Ontology	Ont
Capabilities Statement	Cap		

Table 1. Objects of the Ontology Negotiation Protocol.

Operation	**Abbreviation**	**Operation**	**Abbreviation**
Wait for	Wtg	Return	Ret
Request	Req	Interpret	Int
Receive Request for	Rrf	Interpret Request for	Irf
Receive	Rcd	Evaluate	Evl
Process	Pro	Evolve	Evo
Forward	Fwd		

Table 2. Operations of the Ontology Negotiation Protocol.

	Que	Qre	Ack	Rej	Dcl	Cap	Cfi	Cla	Exr	Exf	Ont
Wtg	x	x	x			x	x	x	x	x	
Req						x	x	x	x	x	
Rrf						x	x	x	x	x	
Rcd	x	x	x	x	x	x	x	x	x	x	
Pro	x										
Fwd	x										
Ret		x	x	x	x	x	x	x	x	x	
Int	x	x				x					
Irf						x					
Evl		x				x					
Evo											x

Table 3. States of the ontology negotiation protocol.

4. Ontology negotiation process

The state machine structure determines the shape of ontology negotiations. It does not determine how an agent decides whether to ask for clarification, or how an agent chooses to clarify a previous message. This is the heart of the ontology negotiation process. The fundamental tasks are interpretation, clarification, relevance evaluation, and ontology evolution

4.1 Interpretation

Interpretation is the process of determining whether a message just received is properly understood. In the testbed, messages are sequences of keywords, and the recipient agent tries to interpret each keyword in turn. First, the agent checks its own ontology to see whether the keyword occurs there. If not, the agent queries WordNet to find synonyms of the keyword. Then it checks the ontology for any of these synonyms. If a synonym is located in the ontology, it represents an interpretation of the keyword. Since WordNet may identify distinct meanings for the keyword (homonyms), each synonym is only a possible interpretation, which must be confirmed by the source of the message. The recipient agent therefore requests a *confirmation of interpretation* from the source agent.

4.2 Clarification

If an agent is not able to interpret some of the keywords of a message it has received, it can decide to proceed anyway (if enough other keywords are understood), or it can request a *clarification* from the source of the message. Given the richness of its database, it would be tempting to invoke WordNet as the primary means of clarifying a keyword. This would be pointless, however, since the recipient agent has already gone to WordNet during the interpretation process.

Instead, the message source draws on the following methods of clarification:

- Locating synonyms in the source agent's ontology
- Providing a complete set of specializations (keyword as the union of its subclasses) from the source's ontology
- Providing a weak generalization from the source's ontology
- Providing a definition in formal logic—in particular, defining the keyword as the conjunction of other keywords

The interfaces to these clarification methods are formulated abstractly in the ontology API. It is up to the ontology to decide how to implement them.

4.3 Relevance analysis

Relevance analysis is the process of evaluating the results of a query against the query itself. In our approach the query is specified via keywords, and the results are documents or URLs that are also described by keywords. So the problem reduces to evaluating how well these sets of keywords match. The evaluation is performed for each URL that is returned. The current implementation computes a relevance measure for each result by accumulating *evidence of relevance* from the result document's keywords. Each keyword of the URL is examined in turn; if evidence of the keyword's relevance to the query can be found, the relevance measure is incremented.

Relevance analysis therefore reduces to the following question: given the set of *query* keywords, and a particular *result* keyword (of one of the returned URLs), what would constitute evidence that the result keyword is relevant to the query? If the result keyword *is* a query keyword, then clearly it is relevant. However, it is desirable to have other criteria since we are assuming an environment where there may not be a single controlling ontology. In the ontology API, the following tests are provided for determining whether keyword A and keyword B are relevant to each other:

- A is a specialization of B
- A and B have a common close generalization ("close" means not too far up the generalization tree)
- A and B have similar meanings (as decided by the ontology)
- A implies B
- The ranges of A and B intersect (i.e., A and B are compatible properties)
- A and B are connected by a series of facts that pair-wise have at least one predicate in common

The precise statement of the last criterion is rather involved; there are several variations, all of which contribute different degrees of relevance.

While the accumulation of relevance over terms is standard practice for general-purpose search engines, recourse to an ontology for semantic tests of the above-listed criteria is not. Discipline-specific engines may well resort to such criteria, and to the extent that they do, our inclusion of the criteria in the API is justified by example. In order to maintain flexibility with different levels of semantic support, the API allows an ontology to implement only a subset of the relevance criteria. In our current testbed we have implemented only the specialization and connectedness tests.

4.4 Ontology evolution

The negotiation process culminates in one or both agents modifying their ontology to introduce a new concept, a new distinction, or simply a new term for an existing concept. As in the case of explaining relevance, the algorithms and/or rules for modifying the ontology will depend on the ontology's representation and the tools used to maintain it. The API specifies only the kinds of updates that may be performed.

5. Example

In a typical scenario, a scientist enters a query to determine whether there is any research on cyclical interactions between global warming and industrial demographics, i.e., whether the geographic effects of warming impact economies in ways that might, in turn, impact climate change. Such an example has abundant potential for multi-disciplinary input and consequent ontology mismatches.

The user agent broadcasts a Request for Capability Statement to determine which agents (representing which archives) might be helpful in satisfying the scientist's request. Some of the archives' agents respond with Capability Statements—essentially, statements of areas of focus. The user agent does not understand some of these responses because the vocabulary is from another (potentially relevant) field.

The user agent consults WordNet to find familiar terminology that is synonymous with the opaque terminology of the Capability Statements. WordNet is of some help but not enough to allow the user agent to evaluate the relevance of the responding agents' statements. In particular, WordNet might provide many synonyms for certain terms, and none for other terms. In those cases in which several alternative meanings were returned, the user agent selects the one that appears most relevant to the scientist's query.

If the user agent has been able to conjecture an interpretation of the Capability Statement, it sends a Request for Confirmation of Interpretation to the archive's agent. If it is not able to interpret some terms in the Capability Statement at all, the user agent issues a Request for Clarification.

In response to a Request for Confirmation of Interpretation, an archive's agent may send a Confirmation or, if the interpretation was inaccurate, a Clarification. The process continues until the user agent decides it has enough information to choose those archives it wants to enlist in responding to the scientist's query. It then forwards the query to one or more archive agents.

Now it is the archive agents' turn to interpret and, if necessary, request confirmation of interpretation or clarification of the query. The same sub-protocol that just occurred at the user agent's initiative may now occur in reverse, this time concerning the scientist's query itself. If an archive's agent decides that it lacks the ability to respond to the query, it may decline. Alternatively, it may try to get other archives to respond. In that case, it forwards the query using the same protocol as the user agent has used (and is using) on behalf of the scientist.

When one or more archives return search results to the user agent, another interpretation process ensues. First the user agent tries to interpret the keywords of the document descriptors in the result set. If there is a problem understanding the descriptors, another interpret-clarify-confirm cycle starts. When the user agent has decided it understands the search result descriptors enough to evaluate their relevance to the scientist's query, it begins the relevance evaluation.

The user agent then looks for evidence that each document is relevant to the query. The user agent consults the scientist's ontology to determine the relationships between the document keywords and the query terms. If a document achieves a relevance score greater than a certain threshold, it is accepted and its descriptor is cached for display to the scientist. If not, the user agent sends a Request for Explanation of Relevance to the archive that located the document.

When an archive's agent receives a Request for Explanation of Relevance, it goes through the same evidence collection process as the user agent did, but with the archive's ontology rather than the scientist's. It then returns the accumulated evidence to the user agent.

The user agent analyzes the Explanation of Relevance to determine three things: first, whether the explanation is credible; second, whether the explanation improves the relevance score of the document; third, if the score has improved, why the user agent was not able to accumulate this evidence itself, using the scientist's ontology.

The credibility of an explanation may depend on facts that the archive's agent accepts as true but the user agent does not (yet). For example, a fact might assert the close interdependency of two phenomena. If the user agent does not recognize the fact as a fact, it may send a Request for Explanation of Fact to the archive's agent. The archive's agent responds with an Explanation of Fact—in effect, an inference trail that summarizes how the fact came to be established in the archive's ontology. If no such record can be retrieved or inferred, the archive's agent can return an empty explanation.

The Explanation of Fact may or may not be acceptable to the user agent. If it is acceptable, the relevance evaluation process continues with the added information obtained from the archive's agent. In addition, the user agent may decide to incorporate the relevance and fact explanations into the scientist's ontology, thus alleviating the need to go through this process again for similar search results.

Finally, when the user agent has filtered out those search results that it deems insufficiently relevant to the query, it presents the remaining results to the scientist. The scientist can then examine the results, access the documents directly through the source archive, or refine the search as with a conventional search engine.

6. Conclusion

We have described a novel approach to attacking the proliferation of information agent ontologies. We have created a software implementation framework that facilitates continued experimentation as well as the development of agents representing other scientific archives. We have layered the reasoning process so that an ontology need only support a well-defined API in order to operate within the framework. We have demonstrated the utility of the approach in a testbed that includes ontology proxies for NASA's Global Change Master Directory and NOAA's Wind and Sea index.

References

[1] Clark, P. and Porter, B. Building Concept Representations from Reusable Components. *AAAI 97*, pages 369 - 376.

[2] Cybenko, G. and Jiang, G. Matching Conflicts: Functional Validation of Agents. *Agent Conflicts: Papers from the 1999 AAAI Workshop*. Technical Report WS-99-08, AAAI Press, Menlo Park, CA, 1999. Pages 14 - 19.

[3] Decker, S., Brickley, D., Saarela, J., and Angele, J. A Query and Inference Service for RDF. *QL'98: The Query Languages Workshop*, Boston, MA 1998. Available at: http://www.w3.org/TandS/QL/QL98/pp/queryservice.html.

[4] Fellbaum, C. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[5] Finin, T., Labrou, Y., and Mayfield, J. KQML as an Agent Communication Language, in Jeff Bradshaw (ed.), *Software Agents*, MIT Press, Cambridge, Available at http://www.cs.umbc.edu/kqml/papers. 1997.

[6] FIPA Agent Communications Language, FIPA 97 Specification, Version 2. Available at: http://fipa.org/repository/index.html. October, 1998.

[7] Guarino N., Masolo C., and Vetere G., OntoSeek: Content-Based Access to the Web, *IEEE Intelligent Systems* 14(3), May/June 1999, pages 70-80.

[8] Heflin, J., Hendler, J., and Luke, S. Coping with Changing Ontologies in a Distributed Environment. *AAAI-99 Workshop on Ontology Management*, AAAI/MIT Press, 1999.

[9] KIF. Knowledge Interchange Format: Draft Proposed American National Standard, NCITS.T2/90-004. Available at: http://logic.stanford.edu/kif/dpans.html . 1998.

[10] McCallum, A., Nigam, K., Rennie, J., and Seymore, K. Building Domain-Specific Search Engines with Machine Learning Techniques. *Intelligent Agents in Cyberspace: Papers from the 1999 AAAI Symposium*, March 22 - 24, Stanford, CA. Technical Report SS-99-03, AAAI Press, Menlo Park, CA, 1999. Pages 28 – 39

[11] Takeda, H., Iwata, M., Sawada, A., and Nishida, T. An Ontology-based Cooperative Environment for Real-world Agents. *Second International Conference on Multi-Agent Systems (ICMAS 96*). AAAI Press, 1996. Pages 353 - 360.

[12] Tarski, A. On Undecidable Statements in Enlarged Systems of Logic and the Concept of Truth. *Journal of Symbolic Logic*, Vol. 4, 1939. Pages 105-112.

[13] Wiederhold, G. Mediators in the architecture of future information systems, *IEEE Computer* 25, 1992.