The Knowledge Engineering Review, Vol. 17:1, 7–19. © 2002, Cambridge University Press
DOI: 10.1017/S0269888902000292 Printed in the United Kingdom

Ontology negotiation between intelligent information agents

SIDNEY C. BAILIN[1] and WALT TRUSZKOWSKI[2]

[1] *Knowledge Evolution, Inc., 1215 17th Street, NW, Suite 101, Washington DC 20036, USA; e-mail: sbailin@kevol.com*
[2] *NASA/Goddard Space Flight Center, Goddard Space Flight Center Code 588, Greenbelt, MD 20771, USA; e-mail: Walt.Truszkowski@gsfc.nasa.gov*

Abstract

This paper describes an approach to *ontology negotiation* between agents supporting intelligent information management. 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.

Our objective is to increase the opportunities for "strange agents" – that is, agents not necessarily developed within the same framework or with the same contextual operating assumptions – to communicate in solving tasks when they encounter each other on the web. In particular, we have focused on information search tasks.

We have developed a protocol that allows agents to discover ontology conflicts and then, through incremental interpretation, clarification and explanation, establish a common basis for communicating with each other. We have implemented this protocol in a set of Java classes that can be added to a variety of agents, irrespective of their underlying ontological assumptions. We have demonstrated the use of the protocol, through this implementation, in a test-bed that includes two large scientific archives: NASA's Global Change Master Directory and NOAA's Wind and Sea Index.

This paper presents an overview of different methods for resolving ontology mismatches and motivates the Ontology Negotiation Protocol (ONP) as a method that addresses some problems with other approaches. Much remains to be done. The protocol must be tested in larger and less familiar contexts (for example, numerous archives that have not been preselected) and it must be extended to accommodate additional forms of clarification and ontology evolution.

1 Problem: cooperation between strange agents

Consider the following scenarios:

You are travelling in a foreign country, and you are lost. You stop a passer-by and try to ask for directions, but you do not speak the language and he does not speak yours. So you gesticulate, draw pictures or otherwise try to find a common, if primitive, substitute language.

Perhaps you do speak the language, but the problem is that your car has broken down in a remote area. The inhabitants of this area are unfamiliar with most of the technology we take for granted. You express your need to have the car repaired, but your interlocutor has never owned a car and has no idea what you are talking about.

You are an engineer whose technical accomplishments are so outstanding that you have been invited to sit on the board of your company. You attend the next meeting. While the individual words are familiar to you, you cannot make head or tail of what the other board members are saying. It is as if they are speaking a different language. Yet you know that they are talking about the same projects that you concern yourself with daily.

Two companies merge, and the project teams that do similar work in each company must now learn to work together. Although the two teams develop the same kind of product, their vocabularies are different. Sometimes they use different terms for the same thing (synonyms), and sometimes they use the same term to mean different things (homonyms). Worse, the criteria that one team uses to guide its decisions do not bear any resemblance to those that the other team uses. Not only are the individual criteria different, but also each team's assumed structure of goals and considerations is completely foreign to the other team. How are they going to work together?

It is not news that cooperating problem-solving systems (whether human or automated) must speak a common language and interpret the language similarly. As the series of sketches above is meant to show, it is not just the vocabulary and syntax that must be shared, but a certain amount of semantics as well. The Knowledge Sharing Initiative (KSI), a programme sponsored by the Defense Advanced Research Projects Agency (DARPA), was organised precisely to address this challenge. One branch of the KSI, Ontolingua, was devoted to facilitating the creation of sharable ontologies (Gruber, 1993). Another branch was devoted to developing the Knowledge Interchange Format (KIF), a common means of speaking about ontologies (KIF, 1998). The third branch led to development of the Knowledge Query Markup Language (KQML), which is a common framework applying ontologies (Finin *et al*., 1998).

The problem we address in this report is this: What happens when agents from different backgrounds – *strange agents* – meet and want to coordinate activities? Do we expect agent cooperation to be contained within the boundaries of pre-established domains for which standard ontologies have been defined? Is this expectation realistic?

The ontology-based approach to enabling such interaction is through systematic, multi-level domain modelling. The more domains that are modelled, and the more such models are explicitly interrelated (through higher-level models), the more agents in these domains will be able to cooperate. Two overall strategies can be undertaken towards this end. The first is to attempt a near-universal model (or model of models), as in the Cyc project described by Lenat and Guha (1990), or the WordNet on-line lexicon (Fellbaum, 1998). Strategically, this is analogous to the creation of Esperanto as a universal language for humans (Cresswell & Hartley, 1992).

The alternative approach is to cultivate pockets of common language and interoperability. This strategy assumes that there will always be cultural variations and therefore boundaries to interoperability. It seeks, however, to facilitate the creation of new cultures (domain models), and the evolution and/or recombination of existing ones, so that new opportunities for cooperation are continually recognised and taken. Maedche and Staab (2001) and Kahng and McLeod (1996) describe investigations of the use of machine learning to support such an evolution process.

Domain modelling is difficult, time-consuming and expensive. In essence, it is a standardisation activity within a technical community. Such standards are analogous to linguistic conventions within human communities. Humans, however, are not entirely bound by these conventions; they have the capacity to explore opportunities for shared meaning across community boundaries, and to modify, extend, or refine existing conventions opportunistically. Our expectations of agents are likely to be frustrated if, when they have become a ubiquitous presence on the web, they do not have a similar capacity.

2 Solutions

Several approaches to resolving heterogeneity have been tried, each with its own strengths and weaknesses. A useful survey of approaches to integrating information on the World Wide Web can be found in Bouguettaya *et al*. (2000). A comparative evaluation of approaches as applied to a specific domain, bio-informatics, is presented in McEntire *et al*. (2000). Ontology negotiation is meant to be yet another approach, which aims to address some of the shortcomings of previous methods. As we will show, in one sense ontology negotiation is not a radical departure from some of the previous methods, and elements of the negotiation process are foreshadowed in the other approaches. In another sense, however, ontology negotiation is based on very different assumptions, as we will attempt to make clear in this paper. First, however, we review the alternative approaches.

2.1 Standardisation of ontologies

The first approach that one would naturally take to resolve the proliferation of distinct ontologies is to develop standard ontologies for particular domains and obtain agreement on using them. There is strong incentive to do this, for example when the domains correspond to specific industries and the costs of ontology mismatches and consequent broken pipelines can be quantified. Examples of such mismatches in industry are described in O'Leary (2000). A description of efforts to standardise ontologies in particular areas of electronic commerce may be found in Fensel (2001). Applications of this approach to geographic information systems are described by Fonseca *et al*. (2000), Fonseca and Egenhofer (1999), and Wariyapola *et al*. (1999).

Standardisation is a time-consuming and labour-intensive process that involves the reconciliation of interests and agendas of multiple stakeholders. Another familiar phenomenon is the often unforeseeable victory of a de facto standard over the official version. Nevertheless, standardisation of ontology has worked in domains that are sufficiently well bounded and in which the balance of motivating versus fragmenting forces is serendipitous.

2.2 Aggregation of ontologies

The next approach that one might logically take is to develop broader ontologies that include the multiplicity of smaller ontologies. This is a particularly sensible approach in cases in which the realisation of a goal involves the participation of experts (whether human or automated) in several distinct domains. In such cases, the problem of ontology mismatch is not so much the danger of misinterpretation as of incomprehensibility. The design of a spacecraft, for example, requires expertise in diverse disciplines including orbital mechanics, electrical engineering, software design and many others. All disciplines have their own ontologies, which are for the most part orthogonal to each other. In such cases, the most direct solution is to provide the expert in one field with access to the vocabulary and definitions of the related fields. This is a straightforward solution and is often sufficient. A system called StepLib, for example, supports the aggregation of ontologies for geographic information systems (Baptista & Kemp, 2000). Altman *et al*. (1999) describe the RiboWeb system, which aggregates ontologies in molecular biology. A domain-independent approach to aggregating ontologies is provided by the SHOE system (Heflin *et al*., 1999).

2.3 Integration of ontologies

In some cases, distinct disciplines may not be entirely orthogonal, i.e. there may be concepts that play an important role in more than one discipline. Aggregating the ontologies may not be sufficient because of difficulties that appear on the boundaries, or in the overlapping areas, between domains. The classical problems of synonymy and homonymy arise at these points. Two disciplines might use distinct names for the same concept. Alternatively, or in addition, they may attribute different properties to the same concept (e.g. physical versus computational properties), or they may assume different rules governing the use of the concept. StepLib, for example, goes beyond simple aggregation to support integration of ontologies when necessary. Ranta *et al*. (1998) describe an approach to ontology integration in product industries. The issues that arise in ontology integration in medical domains are discussed in Gangemi *et al*. (1998).

The work of Noy and Musen (1999) presents another example of the integration approach: their SMART system provides a semi-automatic approach to ontology merging and alignment. Merging is a form of what we are calling aggregation. In ontology *alignment*, however, the original ontologies are supplemented with mappings that link corresponding or related concepts.

Integration of ontologies requires that such circumstances be recognised and the consistency issues addressed. *Context* is one approach to addressing such issues. A concept can be represented in the integrated ontology as being usable in more than one context. Different attribute and rule sets may be specified as applying to different contexts. The level of difficulty entailed by this depends, again, on the mutual independence of the attribute and rule sets or, viewed another way, on the orthogonality of the contexts. If a rule applied in one context has no meaning in another possible context for the same concept, then there is no consistency problem. If, however, the rules applied in different contexts might conflict with each other, then they must be reconciled. Doing so brings the knowledge engineer back to the same kind of analysis involved in the standardisation approach.

2.4 Mediation between ontologies

The approaches already described aim to create a single ontology out of multiple contributing ontologies. One of the drawbacks of doing so is the loss of independent ownership of the source ontologies. This may be considered a good thing, insofar as it ensures coordinated evolution of the agglomerated ontology. It could, however, inhibit the evolution of an ontology representing a contributing industry or domain. Conversely, the evolution of contributing domains can quickly undo the legitimacy of an agglomerated ontology and foster the appearance of new, decentralised ontologies. This would set the stage for a recurrence of ontology mismatches.

The mediator approach arose as a solution to similar problems in the context of heterogeneous databases. With a proliferation of information sources, it is desirable to hide differences in format and organisation from users. Queries should be satisfied through access to multiple sources, as needed, without extra effort by the user. A unifying schema is not necessarily an optimal or even a feasible solution, both for intrinsic reasons of data and for practical reasons of ownership, intended use, location and other factors. Mediators provide an alternative.

A mediator is a piece of software that translates between different schemata or, in our case, different ontologies. A request for information can arrive at a mediator formulated in terms of one particular ontology; the mediator translates it into the terms appropriate to the potential sources of the information, then, when the information is received, the mediator translates it (or the metadata describing it) back into the terms appropriate to the originator of the request.

The central ideas of the mediator approach are described in Wiederhold and Genesereth (1997), Wiederhold (1995) and Campbell and Shapiro (1995). The COntext INterchange (COIN) project developed the idea of mediating not just differences in terms but also differences in context. As described in Bressan *et al*. (2000) and Goh *et al*. (1999), context in COIN is modelled by means of rules. An example of mediation in geographic information systems is the OBSERVER system, which uses assertions in definitional-logic to express and translate ontological information (Mena *et al*., 1996). An application of mediators to scientific repositories in general is described in (Houstis *et al*., 1999). The KRAFT system uses ontology mediators to support the configuration of designs from parts contributed by multiple organisations (Preece *et al*., 1999).

There is considerable flexibility in implementing this approach, including such issues as the location of the mediator, its ownership, and the way in which information sources are chosen by (or for) it. Invariant in the approach is the notion that the mediator has the intelligence to translate between different ontologies. It thereby relieves other entities, whether people or software, of the need for such knowledge.

One immediate benefit of the integration approach is that problems of consistency can be finessed. By translating concept X in ontology A into concept Y in ontology B, the mediator might be saying that Y is the closest equivalent to X. There may be no occasion to evaluate the consistency of the attributes or rules of the respective concepts. Depending on the information requirements, this may be good or bad. It is good, for example, if it leads to the retrieval of information that would not otherwise have been found. It is bad if it leads to retrieval of the wrong information. In information retrieval terms, mediation may increase *recall* at the expense of *precision*.

2.5 Open ontologies

How, then, does one balance the benefits of translation against its potential cost in loss of precision? One approach is to allow the participating entities (whether human or automated) to do the translation themselves, with a level of precision that they themselves decide is appropriate to the task being performed. In order for this to be possible, without recourse to an agglomerated ontology, each participating entity must be able to access the other entities' ontologies. Such access requires either a standard way of representing ontologies or a standard interface to ontologies. Both approaches have been developed. A survey of both mediation and open-access approaches to ontology-sharing in the environmental sciences may be found in Visser *et al*. (2001).

The Open Knowledge Base Connectivity (OKBC) interface is an analogue to the Open Data Base Connectivity (ODBC) standard. It provides a set of calls or queries that can be issued to an ontology management system to learn about the structure and content of an ontology. For example, given a concept in the ontology, an OKBC call can be issued to discover the available specialisations of the concept (Chaudhri *et al*., 1998).

2.6 Distributed open ontologies

OKBC is strictly an interface: it leaves the internal representation of an ontology up to the ontology authoring system. With the advent of the World Wide Web, and the recent movement to enrich web pages with explicit machine-processable semantics, an alternative approach to open ontologies has arisen. This approach is to encode an ontology in a textual web page using a formal language (Fikes & Farquhar, 1999). The most commonly used language of this type is the Resource Description Framework (RDF) defined by the World Wide Web Consortium (W3C) – see, for example, the text by Hjelm (2001). StepLib, the geographic ontology system mentioned above as supporting the aggregation and integration approaches, manifests some properties of the open ontology approach through its use of RDF to encode ontologies.

RDF is intended to be used in conjunction with the eXtensible Markup Language (XML), which is described in Harold and Means (2001). Web pages are given explicit semantics by marking them up with XML tags, which are essentially labels for different types of information. The meaning of the tags is provided by other web pages, which contain RDF definitions of the tags. Alternatives to RDF include the XML-based Ontology exchange Language (XOL) and the Ontology Inference Layer (OIL). The most recent incarnation of this approach is the DARPA Agent Markup Language (DAML) combined with OIL – a combination commonly referred to as DAML + OIL. Although the constructs available in these languages differ, the overall idea of encoding ontology definitions as web pages is common to all of them (Fensel *et al*., 2001).

2.7 Ontology exchange

An alternative to open access is to communicate ontological information directly via an exchange of messages. This eliminates the need for one agent to go chasing after another agent's ontology, which might be distributed in many locations over the Web. It opens the possibility of self-explanatory messages between agents that do not necessarily speak the same language.

There is an obvious trade-off between the convenience of self-contained messages and the size of an ontology. This translates into a trade-off between ontology exchange and the open-access approach. One way of mitigating the problem of ontology size is to include, in the messages exchanged, only those portions of the agents' respective ontologies that are needed for effective communication in the performance of the current task. This idea is one of the foundations of ontology negotiation, which we describe next.

In one sense, there is not much difference between ontology exchange and the distributed open-access approach. Distributed open access may also suffer from the problem of ontology size: an agent must parse an entire web page encoded, for example in RDF or OIL, in order to find the definitions that it needs. We can imagine an approach that lies mid-way between distributed open access and ontology exchange. Messages exchanged between agents could include references to specific locations within (for example) an RDF-encoded ontology. This would keep messages concise while focusing the interpretation process on those definitions that are needed for the current exchange.

2.8 Ontology negotiation

Can one really predict in advance just what definitions are needed, however? The process of translation involves a chain of replacements that ends only when the resulting terms are familiar to the interpreting agent. This might involve resorting to more and more primitive terms until a shared reality is found in which to express the original concepts. Or it might involve successive generalisations until a common concept is found.

When we, as humans, are asked to explain a particular concept, we may respond by referring the querying person to an appropriate text. We have no guarantee, however, that the text will be intelligible to that person. They may have to consult other texts, possibly cited in the text to which we pointed them, in order to understand our citation, and this chain can continue indefinitely.

Ontology negotiation addresses this issue by building on the notion of ontology exchange, but doing so within a protocol that provides for successive clarifications. We have used certain patterns in human conversation as a model for the protocol. In human conversation, we sometimes explain our meaning by referring our conversational partner to a document. More often, however, we respond to proffered interpretations or requests for clarification by directly elaborating on the intended meaning. The heart of the ontology negotiation protocol is this iterative and recursive process of interpretation and clarification.

3 Overview of the protocol

The ontology negotiation protocol is based on a small set of message types, presented in Table 1. *Capability statements* are a way of narrowing the range of potential respondents prior to issuing a query. *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 results of a query. In either case it is up to the querying agent to decide what to do next, e.g. to inform the user or to try another server agent.

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. The relationships between these particular message types are illustrated in Figure 1.

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 subsequent explanation may include *facts* that are true in the server agent's ontology and knowledge base, and these may themselves require explanation (Figure 2).

Table 1 Message types of the ontology negotiation protocol

Request for capability Statement	Declination	Request to confirm interpretation	Request to explain relevance
Capability statement	Acknowledgement	Confirmation of interpretation	Explanation of relevance
Query	Rejection	Request for clarification	Request to explain fact
Query results		Clarification	Explanation of fact

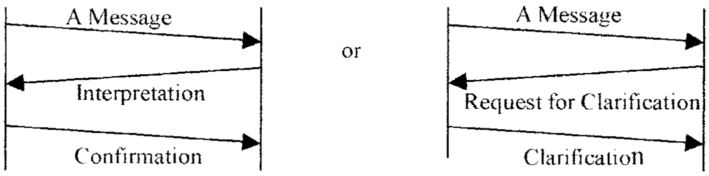

Figure 1 Interpretation and clarification in the ONP. Initial "message" can be query, query results, request for capability statement or capability statement. "Interpretation" is actually a "request to confirm interpretation" in terms of Table 1

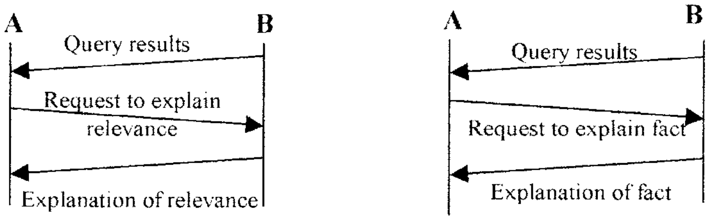

Figure 2 Evaluation and justification of query results

ask-all
:content query ["global warming" industry]
:reply-with User993870550422
:ontology User
:language English
:sender User
:receiver GCMD

Figure 3 Encoding an ONP query in KQML. The italicised terms are KQML parameter keywords

The message types are defined as specialisations of KQML performatives. The protocol establishes additional semantics associated with the specialisations. The query and request types are instances of the KQML *ask-all* performative. The KQML *content* of such a message includes the specific ontology negotiation message type (as in Table 1) and all necessary supporting information (either the query terms or the item for which clarification, confirmation or explanation is sought). Similarly, query results, clarifications, confirmations of interpretation and explanations are encoded with the KQML *tell* performative. The KQML *content* of such a message includes the ontology negotiation message type as well as supporting information such as the query result document keywords, the clarifying terms or a requested explanation. Figure 3 illustrates how a query is encoded in KQML.

As a particular use of these constructs, the ONP represents a commitment to a certain style of communication. The commitment does not, however, interfere with an agent's ontological assumptions or with its patterns of communication outside the realm of ontology negotiation. The objects and operations, while constituting a kind of ontology in their own right, do not usurp the role of an agent's domain ontology. They represent, in effect, an ontology for ontology negotiation. This level of commitment to some basic communication primitives specifically for the purpose of ontology negotiation is, we believe, essential, and should not lessen the benefits provided by the ONP in enabling agents to coordinate cross-domain problem-solving activities.

4 The ontology negotiation process

The heart of the ontology negotiation process consists of the tasks of interpretation, clarification, relevance evaluation and ontology evolution. The basic pattern of interpretation and clarification is shown in Figure 4.

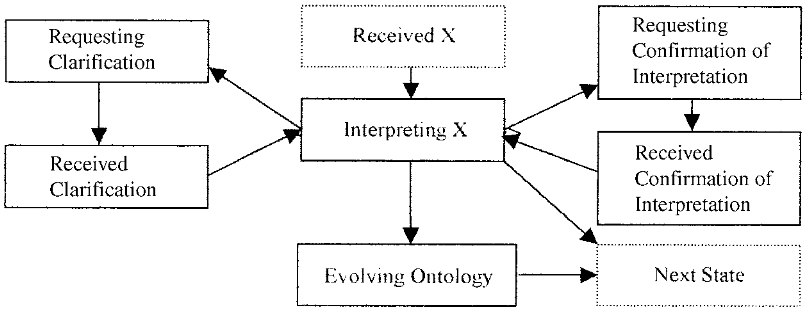

Figure 4 State transitions for message interpretation. "X" represents an arbitrary message that requires interpretation by its recipient

4.1 *Interpretation*

Interpretation is the process of determining whether a message just received is properly understood. In the current implementation, 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 the WordNet lexical database 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:

- locate synonyms in the source agent's ontology,
- provide a set of specialisations or instances from the source's ontology,
- provide a weak generalisation from the source's ontology, and
- provide a definition in formal logic.

The interfaces to these clarification methods are formulated abstractly through an ontology server application program interface (API), whose Java specification is shown in Figure 5. It is up to the ontology server to decide how to implement this interface. The API may be considered a subset of OKBC, and in subsequent versions we intend to make this relationship explicit.

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. This process is performed by the querying agent when it receives the query results, as well as by the archive agent if it receives a request to explain relevance.

```
public interface Ontology {
    // isWordNetOntology prevents duplicate calls to WordNet by client and server
    public boolean isWordNetOntology();

    public String getName();
    public Vector getRelated(String term);
    public boolean isKnown(String term);

    // Clarification methods
    public TermTree getSynonyms(String term);
    public TermTree getConjunctiveDefinition(String term);
    public TermTree getDisjunctiveDefinition(String term);
    public TermTree getDefinition(String term);
    public TermTree getSlightGeneralization(String term);
    public TermTree getImportantCase(String term);

    // Relevance evaluation methods
    public boolean checkSpecializes (String term1, String term2);
    public boolean checkImplies (String term1, String term2);
    public boolean checkGeneralizesALittle (String term1, String term2);
    public boolean checkSimilarMeaning (String term1, String term2);
    public boolean checkRangesIntersect (String term1, String term2);
    public boolean checkConnected (String term1, String term2);

    public TermTree explainSpecializes (String term1, String term2);
    public TermTree explainImplies (String term1, String term2);
    public TermTree explainGeneralizesALittle (String term1, String term2);
    public TermTree explainSimilarMeaning (String term1, String term2);
    public TermTree explainRangesIntersect (String term1, String term2);
    public TermTree explainConnected (String term1, String term2);

    // Evolution methods (inverses of the clarification methods)
    public void putSynonyms(String term, TermTree t);
    public void putConjunctiveDefinition(String term, TermTree t);
    public void putDisjunctiveDefinition(String term, TermTree t);
    public void putDefinition(String term, TermTree t);
    public void putSlightGeneralization(String term, TermTree t);
    public void putImportantCase(String term, TermTree t);

    // Evolution methods (inverses of the relevance evaluation methods)
    public void putSpecializes (String term1, String term2);
    public void putImplies (String term1, String term2);
    public void putGeneralizesALittle (String term1, String term2);
    public void putSimilarMeaning (String term1, String term2);
    public void putRangesIntersect (String term1, String term2);
    public void putConnectedByFacts (String term1, String term2);

}
```

Figure 5 The ontology application program interface

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 (describing, in the query response, 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 server API, the following tests are provided for determining whether keyword A and keyword B are relevant to each other:

- A is a specialisation of B,
- A and B have in common a slight generalisation,
- 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), and
- A and B are connected by a series of facts that pair-wise have at least one predicate in common.

We chose these criteria on the basis of the human conversational model. They are not intended to be canonical in any sense, and we envision the choice criteria evolving as we experiment with the protocol. The API allows an ontology server to support only a subset of the criteria. In the current version of the software, we have implemented only the specialisation 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 new constraints, 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 best we can hope for in a general capability is an API that specifies (logically) the kinds of update that may be performed. As shown in Figure 5, the ontology evolution methods are analogues of the clarification and relevance evaluation (or explanation) methods.

Significant consistency issues arise in the implementation of such methods. For example, a definition offered by agent A to agent B as clarification of a concept may be inconsistent with a definition already established in B's ontology. Depending on the nature of the concepts, the presence or absence of such inconsistencies may not be decidable or tractable. For example, if a first-order theory including arithmetic is being used, then consistency is undecidable.

Currently we sidestep such issues by accommodating alternative interpretations of a term. But this raises a host of questions, not only of how to represent such alternative meanings in an ontology but also, more fundamentally, of how to assess the significance of the difference. For example, does it simply represent a different use of the term, or does it indicate a different set of beliefs about the term's referent? We have conjectured some tentative answers to such questions but have not yet implemented them.

5 A brief scenario

We have experimented using the ontology negotiation protocol in mediating between two scientific databases, the NASA Goddard Space Flight Center's *Global Change Master Directory*, and the National Oceanic and Atmospheric Agency's *Wind and Sea* archive. In a typical scenario, a scientist might enter a query to determine whether there is any research on interactions between global warming and industrial demographics. Such an example has abundant potential for multi-disciplinary input and consequent ontology mismatches. We assume the existence of automated agents acting on behalf of the scientist and the various archives. We assume further that these agents all have the ability to engage in the ONP.

The scientist's agent broadcasts a request for capability statement to determine which scientific 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. Suppose that the scientist's agent does not understand some of these responses because the vocabulary is from another (potentially relevant) field.

The scientist’s agent consults an on-line lexicon to find familiar terminology that is synonymous with the opaque terminology of the capability statements. The WordNet digital lexicon is built into the framework in which we implemented the ONP. The scientist’s agent finds it to be of some help but not enough to evaluate the relevance of the responding agents’ statements. In particular, the lexicon might provide many synonyms for certain terms and none for others. In those cases in which several alternative meanings were returned, the scientist’s agent selects one that appears most relevant to the scientist’s query.

If the scientist’s agent has been able to conjecture an interpretation of the capability statement, it sends a request to confirm interpretation to the archive’s agent. If it is not able to interpret some terms in the capability statement at all, the scientist’s agent issues a request for clarification.

In response to a request to confirm interpretation, an archive’s agent may send a confirmation or, if the interpretation was inaccurate, a clarification. The process continues until the scientist’s 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 the query, and if necessary to request a clarification or confirmation of an interpretation. The same sub-protocol that just occurred at the initiative of the scientist’s agent 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 that the scientist’s agent has used (and is using) on behalf of the scientist.

When an archive returns search results to the scientist’s agent, another interpretation process ensues. First the scientist’s 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 scientist’s agent has decided that it understands the search result descriptors enough to evaluate their relevance to the scientist’s query, it begins a relevance evaluation.

The scientist’s agent then looks for evidence that each identified document is relevant to the query. The scientist’s agent consults its 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 scientist’s agent sends a request to explain relevance to the archive that located the document.

When an archive’s agent receives a request to explain relevance, it goes through the same evidence collection process as the scientist’s portal agent did, but with the archive’s ontology rather than the scientist’s. It then returns the accumulated evidence to the scientist’s agent.

The scientist’s agent analyses 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; and third, if the score has improved, the scientist’s agent tries to determine why it was not able to accumulate this evidence using its own ontology.

The credibility of an explanation may depend on facts that the archive’s agent accepts as true but the scientist’s agent does not (yet). For example, a fact might assert the close interdependency of two phenomena. If the scientist’s agent does not recognise the proffered fact to be true, it may send a request to explain fact to the archive’s agent. The archive’s agent responds with an explanation of fact – in effect, an inference trail that summarises 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 scientist’s agent. If it is acceptable, the relevance evaluation process continues with the added information obtained from the archive’s agent. In addition, the scientist’s agent may decide to incorporate the relevance and fact explanations into its own ontology, thus alleviating the need to go through this process again when similar search results are returned.

Finally, when the scientist’s 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 with the ONP. We have layered the reasoning process so that an ontology server need only support a well-defined API in order to operate within the framework. We have demonstrated the utility of the approach in a test-bed that includes ontology proxies for NASA's Global Change Master Directory and NOAA's Wind and Sea Index.

7 References

Altman, R, Bada, M, Chai, X, Carillo, M, Chen, R and Abernethy, N, 1999, "RiboWeb: an ontology-based system for collaborative molecular biology" *IEEE Intelligent Systems* **14**(5) 68–76.

Baptista, C and Kemp, Z, 2000, "An integrated metamodel for knowledge representation in geolibraries" *Proceedings of the IEEE Advances in Digital Libraries 2000* (ADL 2000) 151–159.

Bouguettaya, A, Benatallah, B, Hendra, L, Ouzzani, M and Beard, J, 2000, "Supporting dynamic interactions among web-based information sources" *IEEE Transactions on Knowledge and Data Engineering* **12**(5) 779–801.

Bressan, S, Goh, C, Levina, N, Madnick, S, Shah, A and Siegel, M, 2000, "Context knowledge representation and reasoning in the context interchange system" *Applied Intelligence* **13**(2) 165–180.

Campbell, A and Shapiro, S, 1995, "Ontological mediation: an overview" in V Benjamins (ed.) *IJCAI Workshop on Basic Ontological Issues for Knowledge Sharing* CEUR Publications and University of Amsterdam.

Chaudhri, V, Farquhar, A, Fikes, R, Karp, P and Rice, K, 1998, "OKBC: a programmatic foundation for knowledge base interoperability" *Proceedings of the Fifteenth National Conference on Artificial Intelligence* 600–607.

Cresswell, J and Hartley, J, 1992, *Esperanto: Teach Yourself* McGraw Hill.

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

Fensel, D, 2001, *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce* Springer.

Fensel, D, van Harmelen, F, Horrocks, I, McGuinness, D and Patel-Schneider, P, 2001, "OIL: an ontology infrastructure for the Semantic Web" *IEEE Intelligent Systems* **16**(2) 38–45.

Fikes, R and Farquhar, A, 1999, "Distributed repositories of highly expressive reusable ontologies" *IEEE Intelligent Systems* **14**(2) 73–79.

Finin, T, Labrou, Y and Mayfield, J, 1997, "KQML as an agent communication language" in J Bradshaw (ed.) *Software Agents* MIT Press; also available at `http://www.cs.umbc.edu/kqml/papers`.

Fonseca, F, Egenhofer, M, Davis, C and Borges, K, 2000, "Ontologies and knowledge sharing in urban GIS" *Computer, Environment, and Urban Systems* **24** 232–251.

Fonseca, F and Egenhofer, M, 1999, "Ontology-driven geographic information systems" *Proceedings of the 7th ACM Symposium on Advances in Geographic Information Systems* 14–19.

Gangemi A, Pisanelli, D and Steve, G, 1998, "Ontology integration: experiences with medical terminologies *Proceedings of the International Conference on Formal Ontologies in Information Systems* (FOIS 98) 163–178.

Goh, C, Bressan, S, Madnick, S and Siegel, M, 1999, "Context interchange: new features and formalisms for the intelligent integration of information" *ACM Transactions on Information Systems* **17**(3) 270–293.

Gruber, TA, 1993, "Translation approach to portable ontologies" *Knowledge Acquisition* **5**(2) 199–220.

Harold, E and Means, W, 2001, *XML in a Nutshell: A Desktop Quick Reference* O'Reilly and Associates.

Heflin, J, Hendler, J and Luke, S, 1999, "Coping with changing ontologies in a distributed environment" Ontology management, Papers from the AAAI Workshop, Technical Report WS-99–13, American Association for Artificial Intelligence (AAAI) Press.

Hjelm, J, 2001, *Creating the Semantic Web with RDF: Professional Developer's Guide* Wiley.

Houstis, C, Nikolaou, C, Lalis, S, Kapidakis, S, Christophides, V, Simon, E and Thomasic, A, 1999, "Towards a next generation of open scientific data repositories and services" *CWI Quarterly* **12**(2) 111–132.

Kahng, J and McLeod, D, 1996, "Dynamic classificational ontologies for discovery in cooperative federated databases" *Proceedings of the First IFCIS International Conference on Cooperative Information Systems* (CoopIS '96) 26–35.

KIF, 1998, "Knowledge interchange format: draft proposed American national standard", NCITS.T2/90–004. Available at: `http://logic.stanford.edu/kif/dpans.html`.

Lenat, D and Guha, R, 1990, *Building Large Knowledge-based Systems: Representation and Inference in the CYC Project* Addison Wesley.

McEntire, R Karp, P, Abernethy, M, Benton, D and Helt, G, 2001, "An evaluation of ontology exchange languages for bioinformatics" *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology* (ISMB 2000) 239–250.

Maedche, A and Staab, S, 2001, "Ontology learning for the semantic web" *IEEE Intelligent Systems* **16**(2) 72–79.

Mena, E, Kashyap, V, Sheth, A and Illarramendi, A, 1996, "OBSERVER: an approach for query processing in global information systems based on interoperation across pre-existing ontologies" *Proceedings of the First IFCIS International Conference on Cooperative Information Systems* (CoopIS '96) 14–25.

Noy, NF and Musen, M, 1999, "An algorithm for merging and aligning ontologies: automation and tool support" Ontology management, Papers from the AAAI Workshop, Technical Report WS-99-13, American Association for Artificial Intelligence.

O'Leary, D, 2000, "Different firms, different ontologies, and no one best ontology" *IEEE Intelligent Systems* **15**(5) 72–78.

Preece, A, Hui, K and Gray, P, 1999, "KRAFT: supporting virtual organisations through knowledge fusion" Artificial Intelligence for Electronic Commerce: Papers from the AAAI-99 Workshop, Technical Report WS-99–01, American Association for Artificial Intelligence.

Ranta, M, Buchner, A, Mantyla, M and Hughes, J, 1998, "Collaboration in heterogeneous environments along product processes" in P. Brunet, C. Hoffman, D Roller (eds), *CAD Tools and Algorithms for Product Design* Universität des Saarlandes.

Visser, U, Stuckenschmidt, H, Wache, H and Vogele, T, 2001, "Using environmental information efficiently: sharing data and knowledge from heterogeneous sources" in C Rautenstrauch and S Patig (eds) *Environmental Information Systems in Industry and Public Administration*, IDEA Group.

Wariyapola, P, Patrikalakis, N, Abrams, S, Elisseeff, P, Robinson, A, Schmidt, H and Streitlien, K, 1999, "Ontology and metadata creation for the Poseidon Distributed Coastal Zone Management System" *Proc. IEEE Advances in Digital Libraries Conference, ADL '99*.

Wiederhold, G, 1996, "Value-added mediation in large-scale information systems" *Database Applications Semantics, Proceedings of the Sixth IFIP TC-2 Working Conference on Data Semantics* (DS-6), 1995, in R Meersman and L Mark (eds) *IFIP Conference Proceedings* 74 Chapman & Hall.

Wiederhold, G and Genesereth, M, 1997, "The conceptual basis for mediation services" *IEEE Expert* **12**(5) 38–47.