AAAI-2000, Seventeenth National Conference on Artificial Intelligence, Austin, TX, July 30-August 3, 2000

DOGGIE: A Multiagent System for Learning Diverse Web Ontologies

Andrew B. Williams
Department of Electrical and Computer Engineering
University of Iowa
Iowa City, IA 52242
abwill@eng.uiowa.edu

Costas Tsatsoulis
Information and Telecommunication Technology Center (ITTC)
Department of Electrical Engineering and Computer Science
University of Kansas
Lawrence, KS 66045
tsatsoul@ittc.ukans.edu

Abstract

The Distributed Ontology Gathering Group Integration Environment (DOGGIE) is a multiagent system that demonstrates how agents with diverse ontologies can locate and translate semantic concepts distributed among them. These agents lack a commitment to a common, pre-defined ontology but share a distributed, collective memory of semantic concepts. In particular, our research addresses the essential ontological diversity of artificial intelligence (Genesereth and Nilsson 1987) which states that any agent can create a conceptualization, or ontology, based on its utility for the task at hand. When agents have diverse ontologies, agent knowledge sharing and communication is made more difficult because of the distributed concept locations, different vocabularies, and disparate references to the same concepts. We present how we address these issues using multiagent learning algorithms and evaluate their feasibility and effectiveness through experimentation using ontologies constructed from Web page bookmark hierarchies.

Keywords: multiagent systems, ontologies

Introduction and Related Work

Much research has been done on agent knowledge sharing through the use of common, pre-defined ontologies. The DARPA Knowledge Sharing Effort (KSE) (Finin, Labrou, and Mayfield 1997) recognized that different knowledge-based systems could not share knowledge between them since they were based on different ontologies. Researchers associated with this effort sought to be able to re-use knowledge bases by creating common ontologies in order to facilitate sharing knowledge. Gruber and Olsen (1994) defined the ontology as the ontological commitments among agents to use a shared vocabulary in a consistent and coherent manner. Although we recognize that sharing and communicating knowledge between agents based on common ontologies facilitates communication, in the real world, agents often have diverse ontologies. Some domains, such as the World Wide Web, may necessitate intelligent agents selfishly inventing ontologies based on their utility for the task at hand. Genesereth and Nilsson (1987) described this situation as the essential ontological diversity of artificial intelligence: any conceptualization of the world is accommodated and is invented by an agent based on its utility. Therefore, there is not always *an a priori* agreement to use the same ontology when agents communicate. Ontolingua's purpose was to provide a common set of ontology description primitives for constructing shared ontologies. However, Mineau (1992) pointed out that this approach of using global ontologies has problems due to the multiple and diverse needs of agents and the evolving nature of ontologies. Gruber (1991) also points out the problem of agreeing on a common ontology when he raises the question on how group consensus can be reached on "what to represent" given that agents have commitments to different tasks, representation tools, and domains.

Research is being conducted in agent communication with *differentiated ontologies*, or concepts that are not shared, but inherit structure from concepts that are shared (Weinstein and Birmingham 1997). Their approach uses rough mapping to identify syntactic and semantic similarity between graphs of concepts with description logic. Unlike most approaches, they allow agents to communicate directly rather than translating to a central, shared language. However, they assume that the unshared terms inherit from terms in shared ontologies while we do not assume our agents use shared ontologies. In general, the ontology problem, or how agents can share meaning, is still considered open (Jennings, Sycara, and Wooldridge 1998).

For a group of agents, the vocabularies for their ontologies may consist of word labels assigned to different semantic concepts. These semantic concepts are classes of instances of the semantic concept described by words in that agent's vocabulary. Thus, the agent's ontology consists of the agent's known semantic concepts, or objects, classes of objects, interrelationships between objects in the world, and the learning and reasoning mechanisms required to learn and share its ontological knowledge.

Approach

Our approach addresses these current weaknesses of agent knowledge sharing by introducing a theory of learning ontologies in a multiagent system with a distributed collective memory. That is, individual agents start off by

Copyright © 1999, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

learning their own ontologies. Then the agents begin to interact with each other to "teach" each other what they know from their own point of view. This is done by first sending out concept-based queries consisting of the concept label and addresses in the collective memory that point to examples of the concept to acquaintance agents. The responding agent acquaintances use their own ontologies to interpret the examples to determine whether they know, may know, or do not know the querying agent's concept. Then the responding agent can send back the query to the original querying agent which can then verify whether or not its acquaintance knows about the same concept. We will describe how this exchange can lead to the querying agent learning agent model knowledge or concept translation knowledge. We describe how our agents in DOGGIE deal with locating distributed concepts, different vocabularies, and disparate references to the same concept.

Distributed Collective Memory

Agents can share a centralized memory, a distributed memory, or a hybrid memory (Garland and Alterman 1996). We define a distributed collective memory (DCM) as a set of base memory objects in the world that can be globally accessed but are selectively stored and conceptualized by agents in the multiagent system. These agents may share this distributed collective memory but create their own individualized ontologies when selectively conceptualizing objects in it. We assume that this distributed collective memory is so large that no single agent has conceptualized every object within it. Since these agents do not share a common, pre-defined ontology there is difficulty when these agents wish to share knowledge.

We observe this phenomenon of a distributed collective memory with agents who possess diverse ontologies in the World Wide Web domain. Agents, representing human users of Web browsers, may seek to locate information on the Web related to their own interests and needs. An individual user can store the addresses of Web pages of interest in a hierarchical bookmark list. The categories of similar Web pages represent semantic concepts that can be learned by the agent representing the human user. The addresses, or URLs, for the Web pages represent the addresses in the distributed collective memory, or the World Wide Web. The instances of the semantic concepts are the Web pages pointed to by their corresponding URLs and describe the semantic concept, or Web page category, individually named by the human user. This hierarchical bookmark list created by the user becomes a type of ontology, or concept taxonomy, based on their own view of the world, or conceptualization. A Web ontology such as Yahoo! or Magellan may consist of thousands of semantic concepts. If a human user were to interpret Web pages to belong to these pre-defined ontologies, it would necessitate having to memorize all of them. This would make this process impractical to most users. Therefore, humans tend to create their own Web ontologies based on their own interests and point of view. This is why a method for learning diverse ontologies in a multiagent system is necessary.

Ontology Learning

Each agent in DOGGIE uses supervised inductive learning to determine a rule-based representation of its ontology. The semantic concepts, or Web page categories, from its ontology, or bookmark hierarchy, are used as the known classes of concepts. The Web pages pointed to by the URL's are examples of the semantic concept instances known by each agent. DOGGIE uses C4.5 (Quinlan 1993) with these concept names and their corresponding examples to determine the semantic concept descriptions (SCD). Some example initial ontology rules, or SCD's, learned by a DOGGIE agent are:

```
(defrule Rule_33 (methods 1) (not (ink 1)) =>
(assert (CONCEPT Comp_CS_Res_Resources)))
; 33 [70.0%]

(defrule Rule_27 (breeders 1) => (assert
(CONCEPT Life_Anim_Pets_Dogs)))
; 47 [75.8%]
```

These semantic concept descriptions resulted from learning an ontology consisting of the *Life:Animals:Pets:Dogs* and *Computer:CS:Research:Resource* concepts from the Magellan (1998, 1999) ontology. The rule pre-conditions contain semantic descriptors for each concept that correspond to discriminating Web page tokens, or boolean features, in the concept examples for an agent. The percentage stored on the line after the rule is the degree of accuracy, or certainty, that the rule will properly interpret unseen examples. The Web pages used for the ontologies are preprocessed to remove stop words and HTML tags prior to the application of the machine learning algorithm.

Discovering Semantic Concepts

Agents in DOGGIE search for similar semantic concepts known in the group by sending concept-based queries to their acquaintances. A concept-based query (CBQ) consists of the agent's concept name, *X*, along with a set of addresses in the distributed collective memory that point to instances that describe the concept. The concept-based query is the mechanism DOGGIE uses to search for the best ontologically near examples (BONEs). The querying agent, Q, sends a KQML message containing the CBQ to some or all of its acquaintances who are asked to respond to the query. The responding agent collects the examples from the DCM, or the Web, tokenizes them, and then seeks to interpret them. The tokens, or semantic descriptors, found at the addresses submitted with the CBQ will trigger some of the rules the responding agent has to describe its own ontology (discussed in the previous section as "semantic concept descriptions".) After all the examples have been interpreted, the responding DOGGIE agent determines whether it *knows (K), may know (M)*, or *does not know (D)* the concept. To determine this, it uses the percentage accuracy value calculated during the initial ontology learning process to determine the *positive interpretation threshold*. After all the examples have been interpreted, the interpretation value is compared with the positive interpretation threshold. If the interpretation value is greater than the positive interpretation value, then the agent *knows* the concept. If the interpretation value is less than the *negative interpretation threshold*, then the agent *does not know* the concept. If the interpretation value is in between these values, then the agent *may know* the concept. When it determines that an example belongs to a particular concept, it keeps track of the frequency and calculates an interpretation value using the following equation:

Equation 1 Interpretation Value

$$I_c = \frac{k_c(x)}{N}$$

, where $k_c(x)$ is equal to one if the interpretation for instance *x* is true for the concept *c* and zero if the interpretation for instance *x* is false for the concept *c* and *N* is the total number of examples.

Disparate References

In a multiagent system, agents with diverse ontologies may refer to the same object with disparate concept names (Bond and Gasser 1988) or they may use the same concept name to mean different things (Huhns and Singh 1998). For example, a teenager may refer to a popular song as “cool”, “hip”, or “da bomb”. In these types of situations, agents with diverse ontologies must be able to learn that they are referring to the same semantic concept even though they have disparate names. For one person, the word “bomb” may refer to a destructive device but in the case of the teenager it refers to how enjoyable, popular, or innovative something is. We have developed a method for the agents with diverse Web ontologies to learn this *concept translation*.

To illustrate how DOGGIE handles this type of situation we use our example Rule_27. The querying agent Q, sends a CBQ for its concept “Man’s Best Friend”. Its acquaintance, R1, receives the query and begins to interpret the examples using its rules, which includes Rule_27. As it finds the token, breeders, in the example Web pages, it calculates the interpretation value according to equation 1. If the R1 agent’s interpretation value is greater than its positive interpretation value for its concept *Life:Animals:Pets:Dogs* then it determines that the Q agent’s concept, *“Man’s Best Friend”*, is the same as its concept, *Life:Animals:Pets:Dogs*. The Q agent can then verify whether or not R1 knows its concept but by a different name. If the Q agent determines that these concepts are the same, it can learn concept translation knowledge. In this example, agent Q learns that agent R1 knows its concept *Life:Animals:Pets:Dogs* as*”Man’s Best Friend”*. This concept translation knowledge is used to direct future CBQ’s in order to improve the communication and the quality of concepts discovered.

Different Vocabularies

Another challenge of diverse ontologies is having agents that possess different vocabularies. Even if agents share the same base language, such as English, they may not all understand or use the same vocabulary. This may cause ambiguity when agents are attempting to interpret examples of a concept from each other. We address this problem by introducing *recursive semantic concept rule learning (RSCRL)*. This algorithm uses an agent’s existing ontology to recursively create new concepts from existing ones in order to learn semantic context rules from semantic concept description descriptors.

When two or more agents wish to share knowledge, they must be able to understand each other in spite of not having exactly the same vocabulary. Although agents may have overlapping vocabularies that are contained in their examples of concepts (e.g. Web page), an agent may be missing a token that may be critical in the interpretation process. From our example Rule_33,

```
(defrule Rule_33 (methods 1) (not (ink 1)) =>
(assert (CONCEPT Comp_CS_Res_Resources)))
  ; 33 [70.0%]
```

we note that one descriptor is the token *methods*. If a responding agent, R, uses this rule to interpret a new concept from a querying agent, Q, that does not have the token *methods*, this can create a problem. Using RSCRL, we can create a pseudo-concept for the token *methods*, learn a semantic context rule for it, and try to re-interpret the CBQ. This creates a new rule for *methods* which can assert the fact that *methods* exists even though the actual token *methods* does not exist. The algorithm for RSCRL can be described as follows:

1. Determine the names of the concepts in the ontology.
2. Create meta-rules for and from the semantic concept descriptions:
   a. Use the meta-rules and the interpreter to find which tokens to learn semantic context rules for, or RSCRL tokens.
   b. Transform the ontology by creating new concepts for these RSCRL tokens.
3. Re-learn the ontology rules.
4. Create the semantic context rules from the semantic concept description rules.
5. Re-interpret the CBQ using the new semantic context rules and the original semantic concept descriptions.
6. Determine whether the concept was verified with the new semantic context rules:
   a. If the concept is verified, learn the applicable agent model or concept translation rules.
   b. If the concept is not verified, recursively learn

the next level of semantic context rules by repeating the above steps if the user-defined maximum recursion depth limit is not reached.

This RSCRL algorithm becomes a type of rule search for rules describing missing descriptor(s) in a semantic concept description. The meta-rules are automatically generated following the following form (for rules with two and three preconditions):

1) If $A \wedge B \Rightarrow$ Concept X
   a) If $\sim A \wedge B \Rightarrow$ Learn semantic context rule for A
   b) If $A \wedge \sim B \Rightarrow$ Learn semantic context rule for B
2) If $A \wedge B \wedge C \Rightarrow$ Concept X
   a) If $\sim A \wedge B \wedge C \Rightarrow$ Learn semantic context rule for A
   b) If $A \wedge \sim B \wedge C \Rightarrow$ Learn semantic context rule for B
   c) If $A \wedge B \wedge \sim C \Rightarrow$ Learn semantic context rule for C

Therefore, using our example Rule_33,

```
(defrule Rule_33 (methods 1) (not (ink 1)) =>
(assert (CONCEPT Comp_CS_Res_Resources)))
 ; 33 [70.0%]
```

the following meta-rule is automatically generated for it during the RSCRL process:

```
(defrule Rule_45 (not (methods 1)) (not (ink 1)) =>
(assert (RSCRL methods)))
```

This meta-rule will flag the agent that the CBQ's example semantic objects do not contain the descriptors *methods* and *ink* and that the agent needs to transform its ontology to learn a pseudo-concept for this descriptor *methods*. This will enable the agent to learn additional ontology rules for this descriptor. Once these RSCRL tokens are determined, the agent searches each ontology concept directory for that token. If the token exists in a concept instance, it is removed from the current semantic object and placed in a concept holder named after the token. This builds up these pseudo-concepts with semantic objects, i.e. Web pages, which contain these tokens. Then using our supervised inductive learning algorithm, we are able to generate additional ontology rules.

The semantic context rule generated for the descriptor *method* is:

```
(defrule Rule_29 (not (methods 1)) (this 1)
(management 1) => (assert (methods 1)))
```

This rule states that for the current CBQ, if the *methods* token does not exist but the tokens *this* and *management* do exist, then we can assert the fact that the *methods* token does exist within the context of the current ontology. This is a novel method for determining whether a descriptor's "meaning" exists given the current vocabulary even though the exact token is not used in the current concept category.

Evaluation and Results

The data used to test and measure DOGGIE consisted of random concept categories taken from the Magellan ontology (1998, 1999). The Magellan ontology consisted of approximately 4,385 nodes, or concept categories. Each of the concepts we used had 20 Web pages in them. Each DOGGIE agent was assigned 5 to 12 concepts for their individual ontologies. We used the Magellan ontology to insure that we could make an objective evaluation of whether or not two concepts were similar. We used 10 instances per concept for our supervised inductive learning and 10 different instances for testing. The agent model rule learning, concept translation learning, and recursive semantic context rule learning (RSCRL) experiments were run in 4-, 8-, and 16-agent configurations. Examples of Magellan ontology concepts used included:

- Arts/Architecture/Resources_and_Professional_Organizations
- Business/Companies/Chemicals_Petrochemicals_and_Pharmaceuticals
- Computing/Internet/For_Net_Novices

We used the JESS inference engine (Friedman-Hill) to implement our CBQ interpreter. When testing our multiagent concept translation algorithm, we gave each concept name a unique name yet still maintained their semantic content by not changing the concept instances assigned to it.

Performance Measurements

Our measure of the group's search performance measurements included the average concept precision and concept recall Concept precision is the ratio of the number of relevant *concepts* retrieved to the total number of *concepts* retrieved:

Concept Precision = # of relevant concept retrieved / # of concepts retrieved.

Concept precision differs slightly from traditional information retrieval (IR) precision since DOGGIE is actually seeking a particular concept name rather than a particular document. A relevant concept is determined if the querying agent verifies that the responding agent's concept falls into the K region when interpreting its response.

Concept recall is the ratio of the number of relevant concepts retrieved to the total number of relevant concepts in the distributed knowledge base:

Concept Recall = # of relevant concepts retrieved / total # of relevant concepts

The total number of relevant concepts is the total number located among *all* the agents in the entire group. The total number of agent model rules, concept translation rules, or concept relation rules learned are used as measurements of group learning.

We ran experiments on DOGGIE to determine the feasibility of multiagent learning of ontologies among a group of agents with diverse ontologies and to determine whether they could improve their group performance for relevant semantic concepts through their collective experience. The agents randomly selected concepts to query and then after sending each of their known concepts once, they sent another iteration of their concepts in random order.

We ran three different types of experiments related to the research issues we identified for learning diverse ontologies: agent model learning, concept translation, and recursive semantic context rule learning. We ran our experiments on up to two Sun workstations running the 2.5.1 and 2.6 versions of Unix. A CORBA name server was run on one of the hosts to enable the agents to locate and communicate with one another.

Results and Discussion

We measured the concept precision and concept recall values for the group averaged over two iterations: the learning phase and the post-learning phase. During the initial learning phase of the experiment, the group sent out queries to all of its neighbors for each concept and learned the location of relevant knowledge in the group. On subsequent queries for a concept, the agents used their group knowledge (e.g. agent model knowledge) to direct the queries. We also calculated the concept precision and concept recall using the following equations:

Equation 2 Calculated Concept Recall

$$r_{calculated} = \frac{M_{actual}}{M_{expected}}$$

where the calculated recall value, $r_{calculated}$, is equal the actual number of group knowledge rules (i.e. agent model rules or concept translation rules) learned, M_{actual}, divided by the number of group model rules, $M_{expected}$, given perfect learning and interpretation.

The calculated precision values were calculated using the following equation:

Equation 3 Calculated Concept Precision

$$p_{calculated} = 1 - \frac{M_{error}}{M_{actual}}$$

where the calculated precision, $p_{calculated}$, is equal to one minus the error. The error is the number of group knowledge rules (i.e. agent model rules or concept translation rules) that should not have been learned, M_{error}, divided by the actual number of group knowledge rules learned, M_{actual}.

To test our concept translation learning algorithm, we ran our DOGGIE multiagent system with agents that had different ontologies with unique concept names. These experiments were run in the same fashion as the agent model learning experiments in 4-, 8-, and 16-agent configurations. These concept translation experiments had slightly better results. We found that the concept precision decrease slightly for the 16-agent experiment. When the agents were unable to process the CBQ's and learn concept translation knowledge in time to send the next iteration of queries, this decreased the performance of the group.

The calculated concept recall and concept precision for the concept translation experiments are in Table 1 below. We see that the calculated precision decreases in the post-learning phases as the number of agents increase but it still remains above 50% which is an observable improvement for the concept precision for the two iterations.

Table 1 Calculated Recall and Precision for Concept Translation Learning

# Agents	Calculated Recall (Learning Phase)	Calculated Precision (Post-Learning Phase)
4	0.057	1.0
8	0.201	0.634
16	0.202	0.56

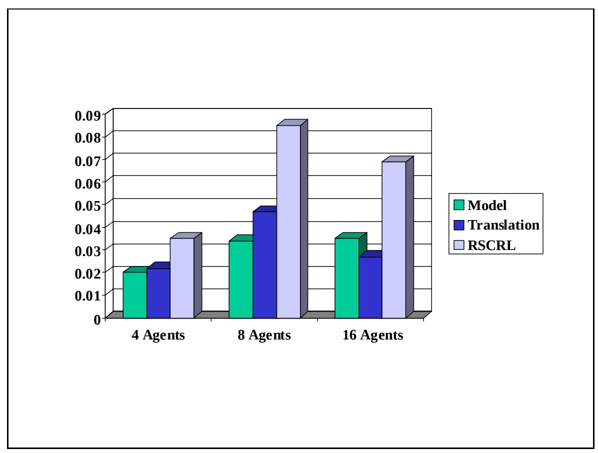
The calculated concept precision and recall for the RSCRL experiments are given in Table 2 below:

Table 2 Calculated Recall and Precision for RSCRL

# Agents	Calculated Recall (Learning Phase)	Calculated Precision (Post-Learning Phase)
4	0.057	1.0
8	0.629	0.818
16	0.148	0.944

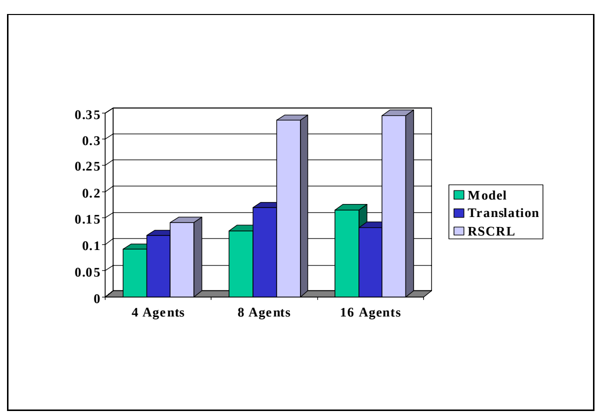
The calculated concept recall and concept precision show an observable improvement over the methods not using RSCRL. We also see how the concept precision for both iterations of concept-based queries improves for RSCRL over the agent modeling and concept translation experiments in Figure 1 below:

Figure 1 Concept Precision



Also we see that the concept recall improves when RSCRL is used for both the agent model learning and concept translation experiments in Figure 2 below:

Figure 2 Concept Recall



Conclusions and Future Work

Our results support the idea that it is feasible to use diverse Web ontologies among agents to learn concept locations and translations among the group. They also show that this learned ontological knowledge can help the group improve its performance in searching for semantic concepts using ontologies constructed from Web page bookmark hierarchies. We found in general that there is a trend towards performance improvement in DOGGIE's concept precision and recall (i.e. the group search performance) as the number of agents increase. Using the recursive semantic context rule learning algorithm improved the average concept over our baseline agent model experiments.

DOGGIE does not specifically address ontology granularity or circularity. However, it does show the feasibility of agents learning how to discover and translate between concepts using an approach that combines agent communication and machine learning. DOGGIE's concept recall and precision performance might be improved in the future by adding more traditional information retrieval techniques such as term frequency and inverse document frequency measures. We also want to see how DOGGIE's performance is affected by having not only diverse ontologies but also diverse learning styles.

References

Bond, A. H., and Gasser, L., Eds. 1988. *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA.

Finin, T., Labrou, Y., and Mayfied, J. 1997. KQML as an Agent Communication Language, In Jeffrey Bradshaw (Ed.), *Software Agents*, MIT Press.

Friedman-Hill, E. 1998. JESS: The Java Expert System Shell, http://herzberg.ca.sandia.gov/jess/ .

Garland, A. and Alterman, R. 1996. Multiagent Learning through Collective Memory, In *Adaptation, Coevolution, and Learning in Multiagent Systems*, Technical Report SS-96-01, AAAI Symposium, Stanford, CA,

Genesereth, M. and Nilsson, N. 1987. *Logical Foundations of Artificial Intelligence*, Palo Alto, CA, Morgan Kauffman Publishers, Inc.

Gruber, T. 1991. The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases, In James A. Allen, Richard Fikes, & Erik Sandewall (Eds.), *Priniciples of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, Cambridge, MA, Morgan Kauffman, 601-602.

Gruber, T. and Olsen, T. 1994. An ontology for engineering mathematics. In J. Doyle, P. Torasso and E. Sandewall (Eds.), Fourth International Conference on Principles of Knowledge Representation and Reasoning, Bonn, Germany, Morgan Kaufmann.

Huhns, M. and Singh, M. 1998. Agents and Multiagent Systems: Themes, Approaches, and Challenges, In *Readings in Agents*, Morgan Kaufmann, San Francisco, CA.

Jennings, N., Sycara, K., and Wooldridge, M. A Roadmap of Agent Research and Development, *Autonomous Agents and Multi-Agent Systems,* 1:7-38, 1998.

Magellan, 1999. http://magellan.mckinley.com.

Magellan, 1998. http://www.lib.ua.edu/maghelp.htm.

Mineau, G. 1992. Sharing Knowledge: Starting with the Integration of Vocabularies, In H. Pfeiffer and T. Nagle (Eds.), *Conceptual Structures: Theory and Implementation, Proceedings of the Seventh Annual Workshop*, Las Cruces, NM, July 8-10, Springer-Verlag, 34-45.

Quinlan, J.R. 1993. *C4.5: Programs for Machine Learning*, San Mateo, CA: Morgan Kaufmann.

Weinstein, P. and Birmingham, W. 1997. Matching Requests for Agent Services with Differentiated Vocabulary, Student Abstract. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Providence, Rhode Island, July 27-31, 850.