REALIZATION OF SEMANTIC SEARCH USING CONCEPT LEARNING AND DOCUMENT ANNOTATION AGENTS

Behrouz H. Far[1] *Cheng Zhong*[1] *Zilan (Nancy) Yang*[1] *Mohsen Afsharchi*[2]

[1] Department of Electrical and Computer
Engineering, Schulich School of Engineering,
University of Calgary, Canada
{far, czhong, zyan}@ucalgary.ca

[2]Department of Electrical and Computer
Engineering, University of Zanjan, Iran
afsharchim@iasbs.ac.ir

ABSTRACT

Currently, search systems are based on commitment to a common ontology. In the real world, it is preferred to enable Web repositories to exchange information freely while keeping their own ontology. This helps contents providers to represent the information independently in the repositories at the expense of bringing complexity to the communication and negotiation. To solve the communication complexity problem we present (1) a method for semantic search supported by ontological concept learning, and (2) a prototype multi-agent system that can handle semantic search while encapsulating complexity of such process from the users. The method introduces a spiral search process and a layered structure of semantic interoperability. Agents, which conduct semantic search on behalf of users, deploy ontologies to organize documents in their corresponding repositories. Through a detailed experiment we will show that agents can improve their search capability by learning new concepts from each other, and consequently, provide better search results to the users.

***Index Terms* —** multi-agent system, semantic search, ontology, concept learning, interoperability, annotation.

1. INTRODUCTION

Current popular search engines are mainly divided into three common categories: horizontal, vertical and combination search engine. Horizontal search features keyword-based indexing and minimal natural language processing. Users need to evaluate the search results for obtaining desired documents. Vertical search indexes content specialized by location, topic, etc., typically tailored to users' preferences. Instead of returning thousands of documents, vertical search engines deliver more relevant results matched with the users' needs. In 2007, Google introduced the "Universal Search" system that replaced some of search results with blended listings that come from vertical sources, such as news, video, images, etc. The blended search engine requires changes like re-categorizing, reorganizing, and/or refining content of documents by grouping them by some attributes. This type of search engines typically work with predefined a ontology.

In contrast with the traditional keyword search technology which depends on the occurrence of words in documents, semantic search denotes one or more concepts in the context of other concepts. Understanding the denotation of concepts can help retrieval part of search engine understand the context of search, the activity the users is trying to perform, thus drive expectations on the categories of documents [5]. The essence of semantic search is *semantic interoperability* towards denotation part in the search phrase. Nowadays, general denotation procedures are realized depending on ontology-oriented means, and ontologies adopted are usually evolved and maintained in a distributed way. Thus, multiplicity of ontologies raises the issue of integration and may lead to ineffective communication among peers involved in a semantic search.

Multi-agent systems (MAS) research offers some solutions for the semantic interoperability. Recently, the idea of having agents *learn* concepts from peers has been suggested as a solution. For example, the work in [7] suggests a method for learning a language and the work in [10] has focused on interactions between two agents to learn a single concept. We have already presented a method for agents to learn concepts from several peers [1, 2] and a method for verification of the learnt concepts [4].

The goal of this research is to devise a process, a model and a prototype multi-agent system (MAS) for semantic search that features concept-learning and semantic interoperability. Research overview and MAS system design will be explained in Sections 2 and 3. A detailed experiment to verify usefulness of the prototype system is provided in Section 4 followed by conclusions in Section 5.

2. RESEARCH OVERVIEW

The general research goals of semantic search using concept learning MAS involves: (1) algorithms for concept learning; (2) methods of concept learning verification; and (3) cooperative search engine and supporting MAS. In this paper we focus on the third goal, by creating a MAS that supports semantic search by taking advantage of concept learning and verification. In order to achieve the goal, the followings objectives must be fulfilled:

1) Individual agents are capable of learning ontological concepts from several peer agents through the interaction with other agents and validating these concepts to better communicate and share information.
2) Semantic search engines are capable of dynamically annotating the data repositories.
3) An integrated method or mechanism is required to support and facilitate the implementation of complex interactions among agents.

To achieve the first objective, ontological heterogeneity in MAS must be solved. This is directly related to the fact that any ontology of certain domain can potentially evolve independently. Therefore the only way for agents with diverse views of the world to understand each other is being able to understand each other's conceptualization of the domain, and then find common grounds among themselves. Previous works on agents' communication mostly assumed a complete common understanding of the concepts used to represent a domain. However, it is now known that this may not be necessarily true. Even if having common conceptualization, still the agents are required to be aware that they have a common conceptualization. This fact is well summarized with the point that any conceptualization of the world is accommodated, and is invented based on its utilization [8]. Consequently, ontology learning solutions are gaining more popularity [9, 10].

To achieve the second objective (i.e. semantic search engine), more advanced than a typical query handling system, we have devised a spiral workflow process that incorporates both concept learning and semantic search (See Figure 1). On one hand, search engines should be capable of responding to the requests according to agreements with concept learning module. On the other hand, annotation procedures of search engine can be done on the fly based on the newly obtained concept instead of fixed predefined ontological concepts.

This is a novel description exposing intrinsic relationship between concept learning and semantic search in a heterogeneous environment. In such environment, concept-learning and semantic search are treated equally as basic roles, involved in the process, which support each other to achieve their own goals by enriching the set of ontological concepts and reducing ambiguity of the search, respectively. Following the spiral process, concept-learning module and semantic search take actions alternately to reach their goals.

To achieve the third objective, the problem of integration and communication between agents raised by multiplicity of ontologies need to be solved. As the essence of semantic search is semantic interoperability among different agents towards denotation part contained in the search expression, semantic search is expected to be able to take advantages of concept learning to establish an integrated mechanism to help find common understandings of concepts, and based on it, higher-level modalities of ontology may accomplish interoperations with respect to those denotations.

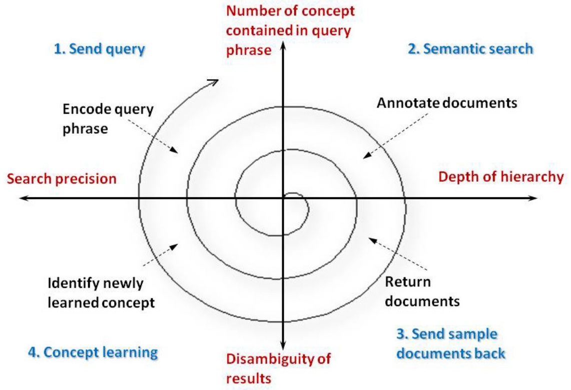

Figure 1. Spiral search and learning process

The work introducing the algorithm for agents to learn concepts from several peer agents (objective 1) has been presented in [1, 2, 12] and a method of verification of concept learning has been presented in [4]. Also the work regarding an initial implementation of a semantic search engine has been presented in [13].

3. SYSTEM DESIGN AND IMPLEMENTATION

Based on the semantic interoperability model [3], we have devised the layered semantic search architecture composed of encoding, lexical, syntactic, semantic and semiotic layers (see Figure 2). According to the definitions of functionalities of layers [13], in order to achieve the interoperations between peers, modeling semantics of concepts and use them in the semantic and semiotic layers need external "schema" (i.e. procedural knowledge), however, for the lexical layer, the declarative contents could solely accomplish modeling by referring concepts to some commonly-understood objects. Considering the fact that the concept learning module [1, 2] is built with a kind of declarative concept learning algorithm, i.e. concentrating on lexical layer, the current implementation of the prototype also has focus on the lexical layer.

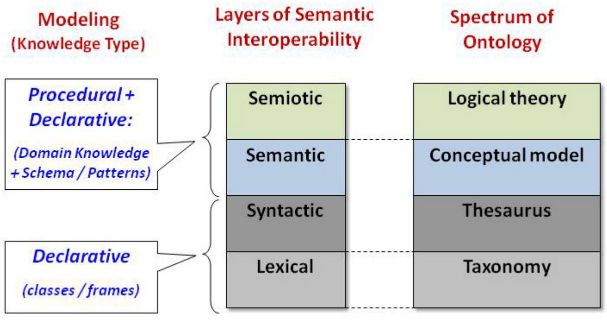

Figure 2. Scope of the prototype system

Besides, from the cognitive sciences perspective, lexical layer would be a basis of communication which ultimately leads to understandings of semantics, so that successfully achieving semantic interoperation would lay a solid foundation for other layers.

Figure 3 shows the architecture for the prototype system. Functional blocks of the system are briefly introduced below.

3.1. Document Annotator (DA)

The documents annotator is used for annotating "molecules", combinations of keyword, on which some well-defined constraints are applied. Creating such annotation, especially dynamically creating annotations is a fundamental role, not only for concept learning, but also for semantic search involving newly learnt concept. Document Annotator is developed using IBM's UIMA (Unstructured Information Management Architecture) [6]. Annotator implements actions *SelectBestConcept*, *SelectPosEx* and *CreateNegEx* according to the UIMA annotation scheme.

3.2. Concept Learner (CL)

The concept learner is responsible for implementing action *Learn* which takes training documents as input and output concept classifier. Also it offers function to do action *Integrate*.

3.3. Communication Engine (CE)

Communication Agent implements actions *QueryConcept* and *ReplyQuery*.

3.4. Personal Assistant Agent (PAA)

Currently, there are two types of PAAs – Training Application (TA) and Semantic Search Application (SSA).

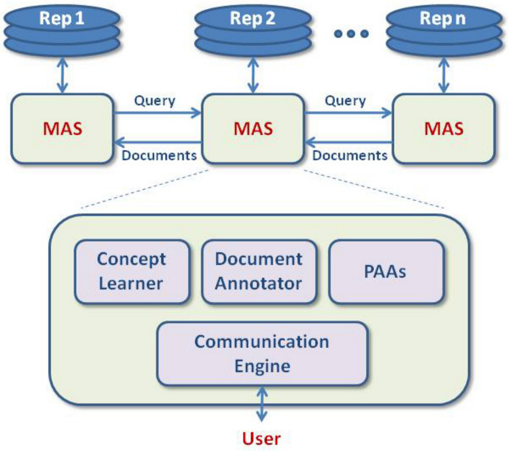

Figure 3. Prototype system and MAS components

GAIA analysis and design methodology [11] is used to design the MAS that implement the above mentioned functions. The MAS associated with each repository is composed of 4 types of agents – Concept Learner, Document Annotator, PAAs and Communication Engine, as shown in Figure 3.

4. EXPERIMENTAION AND EVALUATION

We have designed three experiments using the developed prototype system to observe how the evolution of search results are influenced by concept learning and semantic search and compare them with traditional search. In Experiment 1, a series of traditional queries will be processed by the agents residing on data repositories (Figure 3) in order to observe behaviors of a traditional search and to set benchmarks for comparing with the results of other experiments. Experiment 2 is designed to observe the concept learning stage of the spiral search process. Before sending queries, a new concept is supposed to be identified through interactions between Concept Learner (CL) and Document Annotator (DA) agents, and under the guidance of the attributes of the new concept, the initial repositories are re-structured to be hierarchical repositories. Experiment 3 is carried out using the hierarchical data repositories refined in Experiment 2. It represents the stage of semantic search of the spiral search process. The queries will be processed after the annotation process in which annotators initiatively annotate data repositories they are handling with the same type system which is designed to filter documents.

The disambiguation of search results is measured by a metrics named ROD (Ratio of Disambiguation) which represents the precision of query results.

$$ROD = \frac{Pos}{Pos + Neg} \times 100\%$$

- Pos: number of positive documents. The contents of a positive document meet the query conditions.
- Neg: number of negative documents. A negative document is a false positive document.

The positive or negative is determined a human expert.

4.1. Test data set

The test data set consists of files describing the course syllabi in Computer Science offered by three major universities. A course syllabus file normally contains a course identifier, a course description and the prerequisites of a course. The University of Michigan organizes Computer Science (EECS) as an engineering discipline and as a joint program with electrical engineering; the University of Washington considers Computer Science (CSE) as an engineering discipline but independent from electrical engineering and as a joint program with computer engineering; in Cornell University Computer Science (CS) is a pure science program in the science faculty. The three universities together offer 279 courses in electrical engineering and/or computer science, not including some featureless courses such as seminar course. We set up three data repositories for each university courses, with each repository having a MAS (Figure 3) to handle it. The $\mathcal{A}g_C$,

$\mathcal{A}g_{\mathcal{W}}$ and $\mathcal{A}g_{\mathcal{M}}$ stand for Cornell University, University of Washington, and University of Michigan, respectively.

4.2. Experiment setting

The search goal is to find all courses in *programming languages* from the three data repositories. Query phrases utilized for the three experiments are constructed with five keywords which are related to the search goal. There are five query phrases are listed in Table 1.

Table 1. Query phrases

Query ID	***Feature Content***
F1	Language
F2	Language, Program
F3	Language, Program, Computer
F4	Language, Program, Computer, Science
F5	Language, Program, Computer, Science, Software

4.3. Experiment 1: Traditional search

Traditional search is conducted in Experiment 1. The result is recorded in Table 2, and its visualization of results is represented in Figure 4.

Table 2. Results summary: Experiment 1

	$\mathcal{A}g_{\mathcal{C}}$			$\mathcal{A}g_{\mathcal{M}}$			$\mathcal{A}g_{\mathcal{W}}$		
	Pos.	**Neg.**	**%**	**Pos.**	**Neg.**	**%**	**Pos.**	**Neg.**	**%**
F1	4	1	80	4	16	20	4	15	21
F2	6	2	75	8	30	21	6	28	18
F3	6	5	55	8	55	13	6	50	11
F4	6	7	46	8	56	13	6	51	11
F5	6	7	46	8	62	13	6	55	11

Examining the record of Experiment 1, we can find that the ratio of disambiguation of $\mathcal{A}g_{\mathcal{C}}$ is much higher than the $\mathcal{A}g_{\mathcal{M}}$ and $\mathcal{A}g_{\mathcal{W}}$. we think that this is caused by different composition of data repositories. $\mathcal{A}g_{\mathcal{C}}$ actually holds courses of pure computer science, whereas $\mathcal{A}g_{\mathcal{M}}$ and $\mathcal{A}g_{\mathcal{W}}$ manage courses with composition of both computer science and electrical engineering.

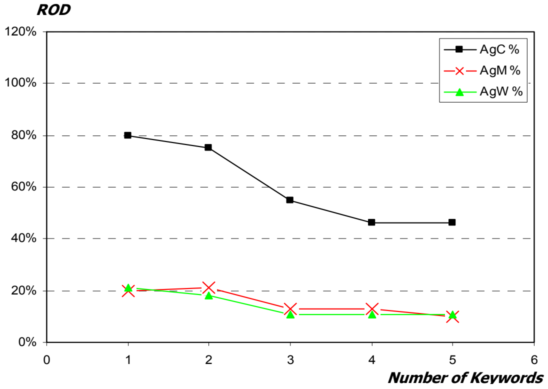

Figure 4. Visualization of results: Experiment 1

In addition, it is worth to mention that:

1) The more terms added to each query, the more documents were retrieved, regardless of whether the documents were positive or negative.
2) Ratios of disambiguation were not guaranteed to be improved with terms added to the query. In this case, it caused the ratios to get worse by adding more terms.
3) All three data repositories were isolated so the number of positive documents was definite. The queries with feature F2 obtained all positive documents in the repositories. After that, no other positive document could be found and search noise made results worse.

From this experiment we can conclude that the composition of data repository influences the search results, confirming that the expected results are significantly correlated with the data repository.

4.4. Experiment 2: Search with Concept Learner

Experiment 2 focuses on examining the behavior of queries, when the Concept Learner has been introduced. The algorithm built into the Concept Learner takes the same data repositories as in Experiment 1 to identify a new concept, *Computer Science*, and then using it to identify all its subcategories. Then using the learnt concept, the data repositories are screened and reorganized for the subcategories of Computer Science [1]. One subcategory, the *programming languages*, is directly adopted to annotate data repositories when annotating action is performed.

Through manipulation of concept learning, the initial flat data repositories were restructured to a two-level hierarchy. We repeated the queries as in Experiment 1 on these structured data repositories. These queries were no longer traditional because at this point any query would have been assumed by the search engine to be a query for all courses of computer science. In practice, new concept (in this case *Computer Science*) will be involved in each query feature to semantically describe it. The results of Experiment 2 are listed in Table 3.

Table 3. Results summary: Experiment 2

	$\mathcal{A}g_{\mathcal{C}}$			$\mathcal{A}g_{\mathcal{M}}$			$\mathcal{A}g_{\mathcal{W}}$		
	Pos.	**Neg.**	**%**	**Pos.**	**Neg.**	**%**	**Pos.**	**Neg.**	**%**
F1	4	0	100	4	4	50	4	6	40
F2	6	0	100	6	10	38	6	13	32
F3	6	3	67	6	13	32	6	19	24
F4	6	4	60	6	13	32	6	19	24
F5	6	4	60	6	20	23	6	19	24

For Experiment 2 we can conclude that:

1. RODs have been improved for all the three repositories and for all the queries. Intuitively, as shown in Figure 5 all the lines representing trends of change of RODs have shifted up significantly.
2. Variations of ROD are still following the same trend as in Experiment 1 (i.e. with the terms added to query, the RODs are decreasing).

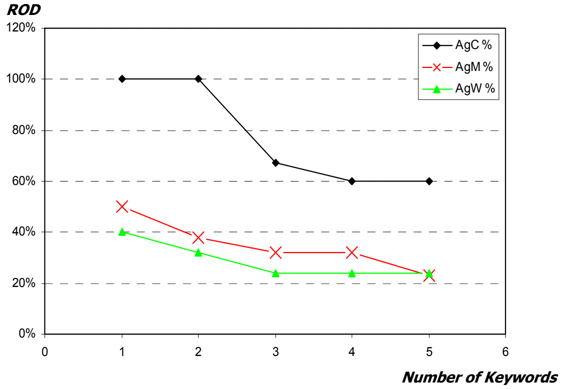

Figure 5. Visualization of results: Experiment 2

The reason for RODs to be different is due to the differences of composition among the data repositories. In the data repositories mixing courses of both disciplines Computer Science and Electrical Engineering such as $\mathcal{A}g_{\mathcal{M}}$ and $\mathcal{A}g_{\mathcal{W}}$, irrelevant courses (e.g. electrical courses) would be eliminated more effectively than that of pure data repository as those of $\mathcal{A}g_{C}$, only holding courses of computer science.

4.5. Experiment 3: Search with document annotation

From the results of Experiment 2, we concluded that through applying concept learner, query performance improved. However, the trends of ROD evolvement remain the same as in Experiment 1.

In this experiment, we apply the Document Annotator (DA) agent to semantically determine if a document is about the searched concept or not, and to see how search performance would be influenced.

Experiment 3 was carried out based on the data repositories refined in Experiment 2 in which the course description documents of computer science were re-categorized under the directory marked by the concept *Computer Science*. At the beginning of the Experiment 3, each data repository was annotated with the same UIMA [6] type system (i.e. kind of concept hierarchy). An aggregate annotator was established consisting of a series of primitive annotators for annotating terms including *language*, *program*, *C*, *C++*, and *Java*. As all the documents to be scanned and relocated, were already under computer science, we were able to replace those non-domain specific terms (computer, software, and science) with those specific terms of the domain computer science (*C*, *C++*, and *Java*). The following expression illustrates a typical annotation logic of the aggregate annotator:

<Language + Program + [C|C++|JAVA] ➔
Computer Programming Course>

This can be interpreted as: "if a three-concept entity created through some logic built in the annotator has been found in the document, then this document was determined to be a target course, i.e. description of *computer programming language*."

Once the annotation process was completed, the corresponding alteration to current data repositories was made. Documents that had not been annotated successfully were removed from the sub-directory dedicated to computer programming language course description. Hence, the ratios of positive documents were raised and the noise that was brought in by adding terms to the query was reduced.

Through the Experiment 2 and the annotation process of the Experiment 3, data repositories were structured with two levels: applying concept learner in Experiment 2 and annotation in Experiment 3. Then we continued to process queries with the same set of features on these restructured repositories. The results of Experiment 3 are listed in Table 4, with visualization in Figure 6.

Table 4. Results Summary: Experiment 3

	$\mathcal{A}g_{C}$			$\mathcal{A}g_{\mathcal{M}}$			$\mathcal{A}g_{\mathcal{W}}$		
	Pos.	**Neg.**	**%**	**Pos.**	**Neg.**	**%**	**Pos.**	**Neg.**	**%**
F1	4	0	100	4	4	50	4	6	40
F2	6	0	100	6	10	38	6	13	32
F3	6	0	100	6	10	38	6	13	32
F4	6	0	100	6	10	38	6	13	32
F5	6	0	100	6	10	38	6	13	32

Compared to Experiment 2, the RODs for the first two queries (i.e. F1 and F2) remained the same as the results of Experiment 2, but the RODs of the rest of queries (with features F3-F5) showed improvement. The reason that the trend lines are more or less horizontal is that adding terms to queries no longer brings noises as in the previous experiments because the sources of noise (i.e., irrelevant documents) have been removed.

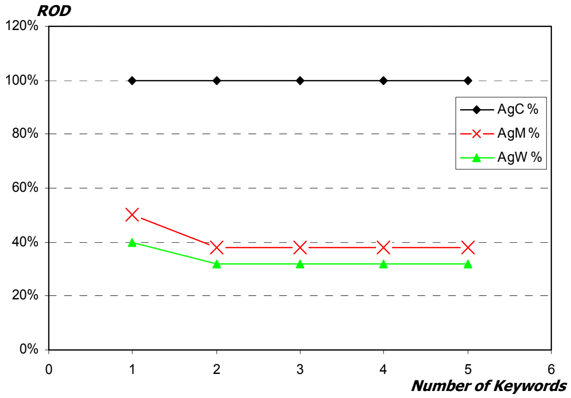

Figure 6. Visualization of results: Experiment 3

4.6. Experiment evaluation and summary

Contribution to the improvement of search results made by both concept learning and annotation is the main concern. Considering that isolated evaluations of either concept learning or annotation would not make sense because the actual working process should be a consecutive generative/spiral process. In order to evaluate contributions made by concept learning and annotation, the percentages of increment of ROD for each query and their average, contributed by concept learning and annotation, are computed respectively. The results are listed in Table 5.

Table 5. Comparison of results

Data Repository	R1(%)	R2(%)	R3(%)	ΔR1(%)	ΔR2(%)
$\mathcal{A}g_C$	80	100	100	25	0
	75	100	100	33	0
	55	67	100	22	49
	46	60	100	30	67
	46	60	100	30	67
	Avg.			**28**	**37**
$\mathcal{A}g_M$	20	50	50	150	0
	21	38	38	85	0
	13	32	38	146	19
	13	32	38	146	19
	13	23	38	77	65
	Avg.			**121**	**21**
$\mathcal{A}g_W$	21	40	40	90	0
	18	32	32	78	0
	11	24	32	118	33
	11	24	32	118	33
	11	24	32	118	33
	Avg.			**121**	**20**

R1: Values of ROD of Experiment 1; R2: Values of ROD of Experiment 2; R3: Values of ROD of Experiment 3; ΔR1: (R2-R1)/R1; ΔR2: (R3-R2)/R2

As the ΔR1 and ΔR2 indicate,

- The average rate of increase of ROD achieved through concept learning on $\mathcal{A}g_C$ (28%) is much less than those on $\mathcal{A}g_M$ (121%) and $\mathcal{A}g_W$ (121%).
- The average rate of increase of ROD achieved through annotation on $\mathcal{A}g_C$ (%37) is larger than those on $\mathcal{A}g_M$ (21%) and $\mathcal{A}g_W$ (20%).
- Both concept learning and annotation made almost identical contributions on data repositories $\mathcal{A}g_M$ and $\mathcal{A}g_W$.

The reason is that $\mathcal{A}g_M$ and $\mathcal{A}g_W$ are mixed data repositories, therefore concept learning had more significant effect on them than on $\mathcal{A}g_C$. However, later in the spiral process, composition of the three data repositories becomes increasingly similar, and consequently, annotation affected the results similarly.

5. CONCLUSIONS

In this paper we presented a method and a prototype MAS for semantic search-learning. This method is based on the architecture of layered semantic interoperability. The central procedure is composed of dynamical document annotation and concept learning mechanisms to solve the problem of semantic heterogeneity in distributed information management with minimum overhead and no need to commit to a common ontology. A detailed experiment was conducted on three data repositories with different ontologies within a specific domain. The experiments were focused on two major parts of the spiral process of semantic search and concept learning. The findings were:

1. When contents of data repositories are relevant to the query keywords, the composition of the data repositories influences the search results. Adding keywords to the query is not helpful for disambiguating the query results.
2. Both Concept Learner (CL) and Document Annotator (DA) agents play significant role in refining the compositions of data repositories in different ways: CL achieves the improvement through reconciling the conflicts of concept between the holders of data repositories, guided by attributes of the newly learned concept. DA, on the other hand, works on its own data repository by applying individual annotation algorithms to restructure the contents.

Future work includes implementation of the role *PeerFinder* which will lead to an open MAS for semantic search.

REFERENCES

[1] M. Afsharchi, B.H. Far and J. Denzinger, “Enhancing Communication with Groups of Agents Using Learnt Non-unanimous Ontology Concepts,” Journal of Web Intelligence and Agent Systems, vol. 3, no. 1-3, pp. 1-16, 2007.

[2] M. Afsharchi, B.H. Far, J. Denzinger, “Ontology Guided Learning to Improve Communication among Groups of Agents,” Proc. AAMAS’06, pp. 923-930, 2006.

[3] J. Euzenat, “Towards a principled approach to semantic interoperability,” A. Gomez-Perez et al (eds.) IJCAI’2001 Workshop on Ontologies and Info Sharing, Seattle, 2001.

[4] B.H. Far, A.H. Elamy, N. Houari and M. Afsharchi, “Adjudicator: A Statistical Approach for Learning Ontology Concepts from Peer Agents,” The 19th Int. Conf. on Software Engineering and Knowledge Engineering (SEKE 07), 2007.

[5] R. Guha, R. McCool, E. Miller, “Using the semantic web: Semantic search,” Proceedings of the WWW’03, 2003.

[6] IBM, “Unstructured Information Management Architecture (UIMA)”, http://domino.research.ibm.com/comm/research_projects.nsf/pages/uima.index.html, 2007.

[7] K.C. Jim, C.L. Giles, “Talking Helps: Evolving Communicating Agents for the Predator-Prey Pursuit Problem,” *Artificial Life 6(3)*, 2000, pp. 237–254.

[8] M.R. Genesereth, and N.J. Nilson, “Logical Foundation of Artificial Intelligence,” Morgan Kauffman Publishers. Inc. Palo Alto. CA, 1987.

[9] L. Steels, “The origins of ontologies and communication conventions in multi-agent systems,” Autonomous Agents and Multi-Agent Systems, 1(2):169–194, 1998.

[10] A.B. Williams, “Learning to Share Meaning in a Multi Agent System, Autonomous Agents and Multi Agent Systems 8(2),” pp. 165–193, 2004.

[11] N. Wooldridge, and D. Kinny, “The GAIA methodology for Agent-Oriented Analysis and Design,” 2000.

[12] Y. Zilan, C. Zhong, B.H. Far, “A Practical Ontology-Based Concept Learning in MAS,” Proc. IEEE CCECE’08, pp. 335-338, 2008.

[13] C. Zhong, Y. Zilan, M. Afsharchi, B.H. Far, “Ontology Learning Supported Semantic Search Using Cooperative Agents,” The 20th Int. Conf. on Software Engineering and Knowledge Engineering (SEKE 08), pp. 123-128, 2008.