5-2007

Radical Agent-based Approach for Intelligence Analysis

Shahram Rahimi
Southern Illinois University Carbondale, rahimi@cs.siu.edu

Henry Hexmoor
Southern Illinois University Carbondale

Bidyut Gupta
Southern Illinois University Carbondale

Recommended Citation
Rahimi, Shahram, Hexmoor, Henry and Gupta, Bidyut. "Radical Agent-based Approach for Intelligence Analysis." (May 2007).

Radical Agent-Based Approach for Intelligence Analysis

Shahram Rahimi, Henry Hexmoor, Bidyut Gupta, Department of Computer Science, Southern Illinois University, Carbondale, IL 62901, {rahimi, hexmoor, bidyut}@cs.siu.edu

Abstract—This paper presents a novel agent-based framework as a decision aid tool for intelligence analysis. This technology extends net-centric information processing and abstraction as well as fusion and multi-source integration strategies. Our information agents traverse and mediate disparate ontologies in different formats providing a foundation for semantic interoperability. The presented system provides knowledge discovery by accessing a large number of information sources in a particular domain and organizing them into a network of information agents. Each agent provides expertise on a specific topic by drawing on relevant information from other information agents in related knowledge domains. Unique advantages include net-centric scalability, principled information assurance, as well as ground breaking knowledge discovery in service of intelligence analysis.

1. INTRODUCTION

There are numerous sources that outline requirements and specifications for intelligence analysis (IA) decision aid tools [2, 6, 14]. Agent orientation is both a natural metaphor for intelligence analysis and an obvious IA panacea. The notion of agents embodies the action orientation we espouse in the Western culture [16]. Most intuitively, the functions of intelligence analysis connote performing *mental actions* of association, correlation, and synthesis, and inference. IA products traditionally yield iconic and acting entities, i.e., agents, such as the AlQaeda. The most notable decision aid systems are the biographical generator Progenie [7] and the role playing game ELICIT [9]. ELICIT allowed human analysts to exchange information and produced offline, empirical validations for group work and collaboration issues. Our main objective is taking up online monitoring, analysis, and proxy function for agents to augment human duties.

An effective knowledge discovery mechanism for IA should provide the foundation for a rich "knowledge space" constructed on top of the basic Internet "data layer." This knowledge layer should be composed of value-added services that process and offer abstracted information and knowledge rather than returning documents (similar to most current web search engines).

To recapitulate, traditional agent-based IA systems have lacked netcentric tenets of facilitating individuals to be disparate in location and in domain. We aim to provide a system that permits collaborations across and within individuals from different communities of interest represented by agents.

Owing to their flexibility and dynamic nature, intelligent agents are widely used as an interface between users and the Internet applications. For example, Bollacker utilized an agent that assists the user in searching for scientific literatures [3], Ackerman developed I-DIAG agent system to refine group collective discussions to be more condensed and strong knowledge [1], and Lieberman employed an agent for helping users browse the Internet by retrieving relevant documents according to users' browsing activities [13]. Other attempts such as an information agent called *SurfAgent* to automatically recommend relevant documents to users based on their profiles [18], or an agent-based brokering facilitation between users and various information resources utilize agents to assist users in retrieving information [17].

More recently, Lesser's agent-based information gathering research resulted in the BIG agent architecture (resource-bounded information gathering) [12]. BIG integrates a number of AI technologies, including a real-time planner and scheduler, a task modeling tool, and an information extraction/understanding component [5, 8]. As another example, BODHI [11] provides a framework for collective data mining tasks on heterogeneous data sources. Agents are distributed to local systems to perform data mining processes, and a centralized coordinator agent is responsible for managing communication and data mining processes among agents.

Furthermore, a formal abstract of the distributed multi-agent system is found in dMARS, which is a successor of procedure reasoning system (PRS) [19]. dMARS is an example of a Belief-Desire-Intention (BDI) approach, which contains four key structures: beliefs, desires, intentions, and plan library. Beliefs correspond to the information related to the world, Desires represent the goal of the tasks, Intentions are chosen desires and will be attempted to be satisfied until either fulfilled or unachievable, and Plan Library specifies the courses of action to achieve intentions. Agents in the dMARS system observe the environmental state, generate possible desires from intentions, and select sequences of plans to achieve these desires.

These approaches are limited to a particular task and are not designed (or scalable enough) to be expanded to a general knowledge discovery system. On the other hand, today's

most advanced multi-agent approaches modestly aim for fragmented information gathering. These architectures do not systematically provide developed knowledge on a target domain that includes all the necessary components to fulfill a query for intelligence analysis. Furthermore, they lack the capability to categorize available information and provide mechanisms to deal with different data representations on the WWW.

2. NETWORK OF INFORMATION AGENTS – THE GENERAL MODEL

We believe that a promising approach to distributed knowledge discovery is to access the large number of information resources by organizing them into a network of *information agents*, as described by Knoblock [31]. The goal of each information agent is to provide information and expertise on a specific topic by drawing on relevant abstracted information from other information agents.
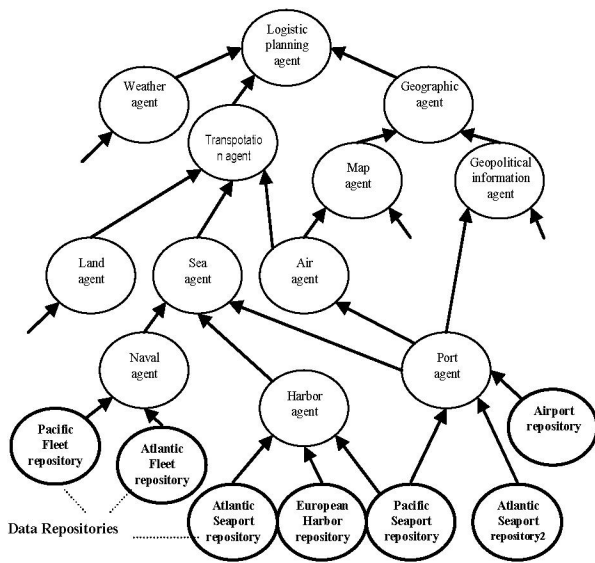

Figure 1 - Network of information gathering agents as viewed by Knoblock

Similar to the way current information resources are independently constructed, information agents are developed and maintained separately. They draw on other information agents and data repositories to provide a new information source that others can build upon in turn. Each information agent is another information source, but provides an abstraction of the other available information resources. An existing information source can be turned into a simple information agent by building a wrapper that allows it to conform to the conventions of the organization. A class of wrapper agents should be built for any given type of information resources (e.g., imagery, graphics, text, formatted text, video and audio, etc.) By simplifying the individual agents to handle one underlying format, it is possible to scale the agent system into networks of agents

with access to many different types of information resources.

Figure 1 provides an example of a logistic planning information network of information agents. There are several points to note about this network that relate to the autonomy of the agents. First, each agent may choose to integrate only those parts of the ontology of its information sources necessary for the task that it is designed for. For example, the transportation_agent may have a fairly complete integration of the sea, land and air_agents, while the logistics_planning_agent may draw on only parts of the knowledge of the weather and geographic_agents. Secondly, we may need to build new agents if we cannot find an existing one that contains all the information needed. For example, if the geographic_agent does not include some particular geopolitical facts required by the logistics_planning_agent, the latter may obtain them directly from the geopolitical_information_agent. However, if much of the information is not represented, an alternative geographic agent would need to be constructed and linked. Thirdly, the network forms a directed acyclic graph rather than a tree because a particular agent may provide information to other agents that focus on different aspects of its expertise (i.e., when the port_agent is accessed by the geopolitical, air and sea-agents respectively). Nevertheless, cycles should be avoided; otherwise, a query may loop endlessly without finding an agent that can actually address it.

3. THE SYSTEM ARCHITECTURE – DETAILED APPROACH

The network of information gathering agents is developed on an autonomous administrative infrastructure (AAI). For this structure, the initial framework design incorporates the use of mobile request agent (MRA), mobile supporting agent (MSA), user interface agent (UIA), information manager agent (IMA), and agent administrator (AA).

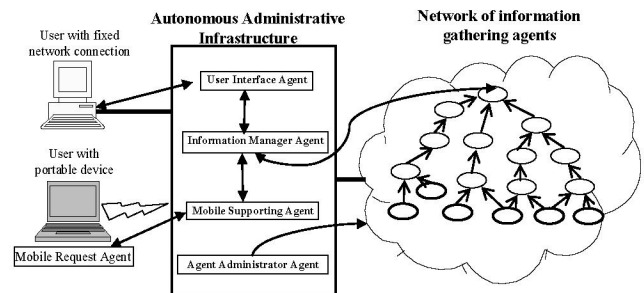**Figure 2** - General infrastructure of the complete knowledge discovery system

To understand how the autonomous administrative infrastructure works, let us consider a simple scenario depicted in Figure 2. A user initiates a query by contacting either a UIA (for users with a fixed connection) or a MRA (for users with a portable device). The MRA is a mobile agent located on the portable device, which migrates to the

host of MSA and provides it with the user request. The MRA and MSA compensate for the limited capabilities of mobile devices and provide an entry point for users with mobile devices to submit requests without the need of high communication traffics. Next, the MSA delegates the request to an IMA (there may be several agents of this type in AAI for multiple concurrent requests), which in turn contacts the appropriate information agent in the network by referring to its directory of domain models (described below). The information agent uses its network, cooperating with other agents in the network (as was described in previous section), to construct and send the real-time result back to the IMA. The IMA applies extra formatting of the information and transmits it back to the MSA or UIA to be presented to the user. In the case of portable device users, MSA provides MRA with functionalities to search and retrieve the requests back to the portable device. The Agent Administrator (AA) in AAI is used for maintaining information agents and their networks. Through AA, the administrator of the system can generate or modify information agents and links among agents to change the topology of the system. The administrator can also introduce new application domains to the system using this agent.

A similar administrative infrastructure for a multi-agent system for geospatial information gathering and integration was successfully designed and developed by one of the authors among others, funded by the DoD-NIMA-NURI grant [15].

The Knowledge of an Information Agent

Each information agent is specialized to a single application domain and provides access to the available information sources within that domain. Each agent contains the *ontology* of its domain of expertise—its *domain model*—and models of the other agents that can provide relevant information—its *information source models*. The domain model is an ontology that represents the domain of interest for the agent and contains the terminologies for agents to interact with each other. The information source models describe both the content of the information sources and their relationship to the domain. These models do not need to contain a complete description of other agents, but rather only those portions that are directly relevant [29]. A model provides a *semantic description* of the domain, which is used extensively for processing queries.

Each information source model has two main parts. First, there is the description of the contents of the information sources. This is comprised of the concepts of interest in terms of the ontology available in other information sources. The terms in the ontology provide the language that is necessary to communicate information sources. Second, the relationship between these information source concepts and the concepts in the domain model are constructed. These mappings are used for transforming a domain model query into a set of queries for the appropriate information sources.

Figure 3 illustrates a fragment of the domain model of the sea-agent that belongs to the organization of Figure 1. The nodes represent concepts (i.e., classes of objects), the thick arrows represent sub-correspondences (i.e., subclass relationships), and the thin arrows represent roles among concepts (i.e., relationships between classes). Some concepts that specify the range of roles have been left out of the figure for clarity. Some are simple types, such as strings or numbers (such as ship-name), while others are defined concepts (such as geoloc-code).
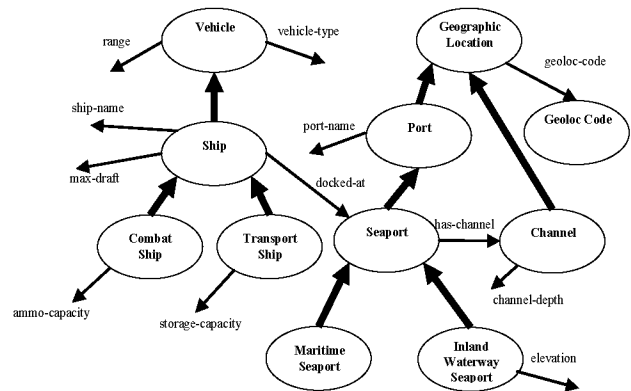

Figure 3 - Fragment of the Domain Model of the Sea Agent
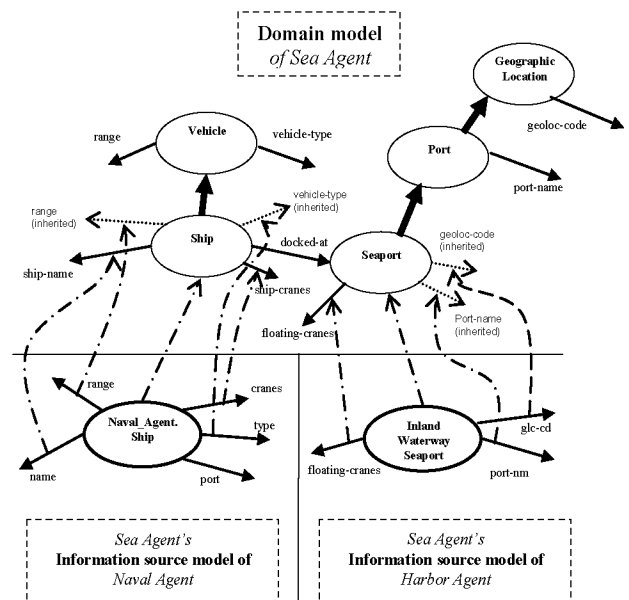

Figure 4 - The Relationship between an Information Source Model and a Domain Model (in the Sea Agent)

Figure 4 demonstrate how an information source is modeled and how it is related to the domain model. All of the concepts and roles in the information source model are mapped to concepts and roles in the domain model. A mapping link between two concepts or roles (dashed lines in the figure) indicates that they represent the same class of information; more precisely, their extensions are equivalent.

Thus, if the user (of the sea_agent) requests all seaports, that information can be retrieved from the Harbor concept of the harbor_agent. Note that the domain model may include relationships that involve concepts coming from different agents (like the role docked-at of the ship concept) but are not explicitly present in any one information source.

Creating ontologies for the information agents to represent their domain knowledge and information source knowledge is fundamental as these ontologies serve as repositories for the agents. Web Ontological Language (OWL) [23] is an ontological language that is utilized in our system. Editors such as SWOOP [24] may be used for OWL to create ontologies as well as to write queries to retrieve information from them. Figure 5 is a screenshot of an OWL class tree created for the system using SWOOP.
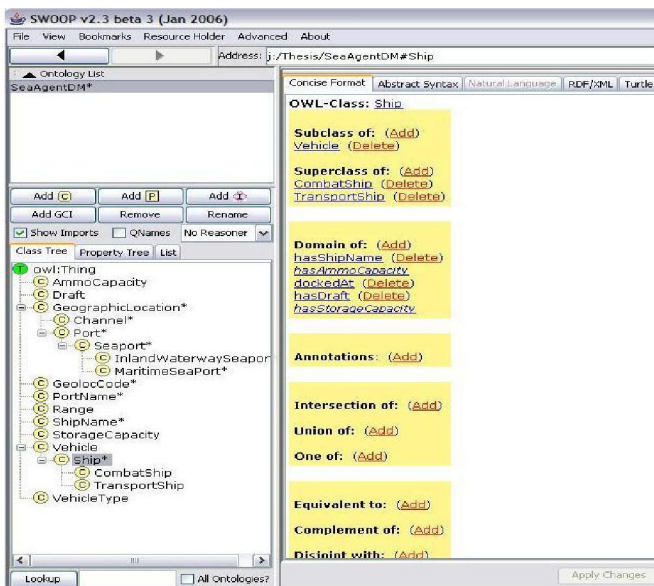

Figure 5 – A snapshot of SWOOP ontology development using editor

In our system, OWL is chosen over some of the other existing languages such as RDF (Resource Description Framework) [4], OIL (Ontology Interchange Language) and DAML (Darpa Agent MarkUp Language) [25]. OWL was selected mainly because of its comprehensiveness. Furthermore, every OWL document is also an RDF document. As such OWL provides the majority of the features provided by these other languages while adding new tags to express relations such as transitivity, equivalence, inverse etc. OWL has three sub-languages that are OWL-Lite, OWL-DL and OWL-Full. While OWL-Full provides high expressiveness of concepts with a large language vocabulary, OWL-DL supports automated reasoning as it is based on Description Logic. Since automated reasoning is crucial in our system OWL-DL was selected. Reasoners such as Pellet [26] and Racer [27] that work with OWL-DL have been evaluated. Currently, Pellet is utilized; however, the system architecture and design allow us to alter this choice if needed.

The Query Mechanism

Whenever an information request is issued, a list of agents that could be related to the subject of the query is produced by an information manager agent (described above). Information agents in this list will initially query their knowledge bases, which are ontological documents to retrieve information. Agents use query languages such as RDQL [20] and SPARQL [21] to retrieve information from their ontological bases. A RDQL query is similar to a SQL query with fields for documents from where information is to be retrieved, conditions that need to be matched for the query and restriction that should be considered. Figure 6 shows the syntax of an RDQL query similar to what is used in our system.

```
SELECT  variable
FROM    document (optional may also be
                           associated by
                the execution environment)
WHERE   condition
AND     Restriction
```

Figure 6 – A sample syntax of an RDQL query

If the required information cannot be found by a requested information agent, it sends information requests down to the agents lower in the hierarchy. In our system, the communication among the agents is done using the Knowledge Query and Manipulation Language (KQML) [22]. Figure 7 presents a sample KQML query generated by a weather agent and sent to a temperature agent asking for the current temperature value at Carbondale, Illinois. Whenever an ambiguous response or an unknown response is received by an agent as a reply, mediation is required. Ontology Mediation is discussed in the next subsection.

```
Ask about
: Content "(Temparature,Carbondale)"
: Language OWL-DL
: Ontology Weather Ontology
: Reply-with "CurrentTemparature"
: Sender Weather Agent
: Receiver Temparature Agent
```

Figure 7 - A sample KQML query

Ontology Mediation

As systems such as the one described in the preceding sections grow and expand, communication becomes more difficult. What begins as a small, controlled experiment becomes a heterogeneous chaos of languages and dialects (described here as ontologies) as newer and different agents are added to the system. Rather than requiring all novice agents to be backwards compatible with existing agents, there needs to be some method of allowing agents with different ontologies to co-exist and interact. Ontological mediation is the ultimate goal for these systems [30].

To develop the mediation algorithm, KQML has been utilized as the agent communication language of the system.

KQML is applied by the agents to communicate with one another and with the *mediation agent* for the purpose of ontology mediation and to resolve ambiguity.

The mediation agent is actually another information agent which is a higher level agent in the hierarchy compared to the mediated agents (i.e., in Figure 1, see_agent could be a mediation agent for a naval_agent and a harbor_agent). To answer to a mediation request, a mediation agent first uses a voting algorithm for arbitration. It initially sends a send-information-agents-list message to the information manager agent (described earlier). Other than the task described above, information manager agent performs two tasks. It performs ontological verification and consistency checking and maintains a list of information agents and brief information about their ontologies so that when required it can provide a directory of agents capable of replying to a particular query. When the list arrives, a multicast message is sent by the mediation agent to all the agents having information about the subject of the mediation. Using a simple counter, votes are collected for the different interpretations and the interpretation receiving higher than a threshold number of votes (currently 66%) is selected as the correct interpretation.

In cases where no clear interpretation emerges from voting, the questioning method is applied (Figure 8). KQML performatives are used to question agents regarding their domain of use, classification, properties, cardinalities, and restrictions among others as needed. Any ambiguity or unfamiliarity that cannot be resolved over a determined period of time is reported to human experts through the Human Aid Agent.
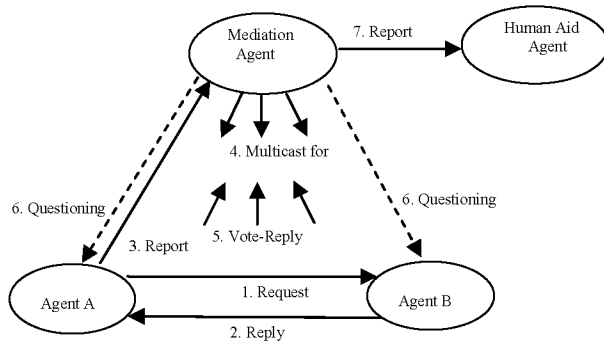

Figure 8 - Illustration of the Mediation Process

4. CONCLUSIONS

Information sources for intelligence analysis are growing exponentially while becoming more diverse and more technically capable. Because of the fast-advancing information technology, intelligence analysts and decision makers are facing increasingly more complex and detailed questions. To answer these questions, they need to be multidisciplinary experts or have access to experts in many specialized disciplines. They may also need to query multiple experts in a particular domain to solve a problem.

The proposed system models the above relationship by using networks of intelligent agents and distributing the knowledge discovery process, on a particular domain, among multiple agents responsible for different sub-domains.

This work has significant scientifical and educational contributions. The presented architecture can be utilized by homeland security as a supporting tool for knowledge discovery for the purpose of intelligent analysis. The proposed model will be fully compatible with our intelligent database agents for geospatial knowledge collection and integration architecture [15], previously funded by a NIMA-NURI in 1999. We are using our previously designed geospatial knowledge collection and integration system to provide our information agents with vector map information through a wrapper agent.

An additional motivation for this work is to provide an architecture that will allow different organizations related to homeland security to have the technological capability for providing information visualization tools such as Starlight (developed by Pacific Northwest National Laboratory [28]) with real-time domain-based information.

REFERENCES

[1] M.S. Ackerman, K. DeMaagd, S. Cotterill, and A. Swenson, "I-DIAG: From Community Discussion to Knowledge Distillation," *AAAI Spring Symposium on Agent Mediated Knowledge Management*, AAAI Technical Report SS-03-01, 2003.

[2] R.V. Balamente, and F. Greitzer, Top Ten needs for intelligence analysis tool development, In 2006 *Proc. of National conference on Intelligence Analysis*, Mitre Corp.

[3] K. D. Bollacker, S. Lawrence, and C.L. Giles, "Discovering Relevant Scientific Literature on the Web," *IEEE Intelligent Systems* 15(2): 42-47, 2000

[4] T. Bray, RDF and Metadata. Retrieved from the Web, Jan 5 2007. http://www.xml.com/pub/a/98/06/rdf.html, 1998.

[5] N. Carver, and V. Lesser. A planner for the control of problem-solving systems," *IEEE Transactions on Systems, Man, and Cybernetics*, Special Issue on Planning, Scheduling and Control, vol. 23, # 6, pp. 1519-1536, 2993.

[6] Robert Clark. *Intelligence Analysis: A Target-centric Approach*, CQ Press, 2006.

[7] P.A. Duboue, K. McKeown, V. Hatzivassiloglou, "PROGENIE: Biographical descriptions for Intelligence Analysis," in *Proceedings of the NSF/NIJ Symposium on Intelligence and Security Informatics*, IEEE, 2003.

[8] D. Fisher, S. Soderland, J. McCarthy, F. Feng, and W. Lehnert. "Description of the UMass systems as used for MUC-6," *Proceedings of the 6th Message Understanding conference*, 1996.

[9] ELICIT. *Experiments in Command and Control within Edge Organizations*, Parity Communications, Inc., 2006.

[10] C.L. Hu, and W. E. Chen. "Mobile agents collaboration for information gathering," *Workshop on Distributed System Technologies & Applications*, NCKU, R.O.C, pp. 537-546.

[11] H. Kargupta, B. Park, D. Hershberger, and E. Johnson, "Advances in Distributed and Parallel Knowledge Discovery," Chapter 5, *Collective Data Mining: A New Perspective Toward Distributed Data Mining*. AAAI/MIT Press, 2000.

[12] V. Lesser, B. Horling, F. Klassner, A. Raja, T. Wagner, and S. Zhang, "BIG: An agent for resource-bounded information gathering and decision making," *Artificial ntelligence Journal*, Special Issue on Internet Information Agents, vol. 118, # 1-2, pp. 197-44, 2000.

[13] H. Lieberman, C. Fry, and L. Weitzman, "Exploring the Web with reconnaissance agents," *Communications of the ACM*, v.44 n.8, p.69-75, 2001.

[14] E. S. Patterson, E. M. Roth, and D. D. Woods, "Predicting Vulnerabilities in Computer-Supported Inferential Analysis Under Data Overload," In *Cognition, Technology & Work*, 3, 224-237, 2001.

[15] S. Rahimi, M. Cobb, M., D. Ali, M. Paprzycki, and F. Petry, "A knowledge-based multi-agent system for geospatial data conflation," *Journal of Geographic Information and Decision Analysis*, vol. 6, No. 2, pp 67-81, 2002.

[16] P. Sengers, J. Kaye, K. Boehner, J. Fairbank, J. Gay, Y. Medynskiy, and S. Wyche, "Culturally Embedded Computing," *Pervasive Computing*, Vol 3, No 1, 2004.

[17] K.M. Sim, and R. Chan. "A Brokering Protocol for Agent-Based E-Commerce," *IEEE Transactions on Systems, Man, and Cybernetics*—part C: Applications and Reviews, vol. 30, no. 4, 2000.

[18] G. L. Somlo, G. L., and A. E. Howe. "Using Web Helper Agent Profiles in Query Generation," In *Proceedings of Autonomous Agents and Multi-Agent Systems* (AAMAS) 2003 Conference, Melbourne, Australia, 2003.

[19] M. D'Inverno, M. Luck, M. Georgeff, D. Kinny, and M. Wooldridge, "The dMARS Architecture: A Specification of the Distributed Multi-Agent Reasoning System," *Autonomous Agents and Multi-Agent Systems,* vol. 9, pp. 5-53, 2004.

[20] A. Seaborne, "RDQL - A Query Language for RDF. Retrieved from the Web," December 4, 2006. http://www.w3.org/Submission/RDQL/.

[21] E. Prud'hommeaux, and A. Seaborne, "SPARQL Query Language for RDF," Retrieved from the Web, December 4, 2006 http://www.w3.org/TR/rdf-sparql-query/, 2006.

[22] T. Finin, T., Y. Labrou, Y., and J. Mayfield, "KQML as an agent communication Language," In *Proceedings of the 3'rd International Conference on Information and Knowledge Management* (CIKM'94), 1995.

[23] D.L. McGuinness, and F.V. Harmelen, "OWL Web Ontology Language Overview," In *World Wide Web Consortium*, Recommendation, 2004.

[24] A. Kalyanpur, B. Parsia, E. Sirin, B. Cuenca-Grau, and J. Hendler. Swoop: A 'Web' Ontology Editing Browser, Journal of Web Semantics Vol 4(2), 2005.

[25] D. Connolly, F.V. Harmelen, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, and L.A. Stein, DAML+OIL March 2001Reference Description, Retrieved from the Web in Jan 2007.

[26] E. Sirin, B. Parsai, P. B. Grau, A. Kalyanpur, and Y. Katz (To Appear, 2006), "Pellet: A practical OWL-DL reasoner," *Journal of Web Semantics.*

[27] V. Haarslev, and R. Moller, "Racer: An OWL Reasoning Agent for the Semantic Web," In *Proceedings of the International Workshop on Applications, Products and Services of Web-based Support Systems*, in conjunction with the 2003 IEEE/WIC International Conference on Web Intelligence, 2003.

[28] Hal Reid, *Starlight Overview and Interview with Battelle's Brian Kritzstein*, Directions Magazine, March 08, 2005.

[29] S. Rahimi, N. Carver, F. Petry, "A Multi-Agent Architecture for Distributed Domain-Specific Information Integration," *Net-Centric Approaches to Intelligence and National Security*, Edited by Roy Ladner, Springer, pp. 129-148, 2005.

[30] H. Sabaa, H. Hexmoor, Towards Semantically Coherent Collaboration, In 2006 *Collaborative Technologies Symposium*, Las Vegas, 2006.

[31] C. A. Knoblock, and J. L. Ambite, "Agents for Information Gathering," *Software Agents*, J. Bradshaw, ed., MIT Press, 1997.