1-2005

A Multi-Agent Architecture for Distributed Domain-Specific Information Integration

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

Norman F. Carver
Southern Illinois University Carbondale

Published in Rahimi, S., & Carver, N. (2005). A multi-agent architecture for distributed domain-specific information integration. Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005. HICSS '05, 113b. doi: 10.1109/HICSS.2005.27

A Multi-Agent Architecture for Distributed Domain-Specific Information Integration

Shahram Rahimi and Norman F. Carver
Department of Computer Science
Southern Illinois University
Carbondale, IL 62901-4511
rahimi@cs.siu.edu

Abstract

On both the public Internet and private Intranets, there is a vast amount of data available that is owned and maintained by different organizations, distributed all around the world. These data resources are rich and recent; however, information gathering and knowledge discovery from them, in a particular knowledge domain, confronts major difficulties. The objective of this article is to introduce an autonomous methodology to provide for domain-specific information gathering and integration from multiple distributed sources.

1. Introduction and Background

The Internet has drastically changed the availability of electronically accessible information. According to a Cyveillance study released in July 2000 [28], the Internet contains over 2 billion unique, publicly accessible pages which are accessed and updated by millions of users internationally. Automated data collection tools and mature database technology lead to tremendous amounts of data stored in databases, data warehouses and other information repositories. The available data sources include traditional and object oriented databases, knowledge bases, flat files, formatted files (such as XML), vector maps and raster images, videos and audios. With the growing number of information sources available, the problem of exploiting and integrating distributed and heterogeneous information sources is becoming more and more critical. Information gathering and integration from Internet and Intranet sources faces several challenges. First, the variety and amount of the data sources are increasing dramatically day by day. Second the general information is unorganized, imprecise, of diverse format, and is distributed on several servers through heterogeneous networks all over the world. Third, the availability and reliability of information are changing constantly. Consequently, in large-scale network environments such as the Internet, it is becoming more and more difficult to use the traditional methods to retrieve and integrate information efficiently and even more difficult to perform knowledge discovery.

An effective information integration mechanism should provide the basis for a rich "knowledge space" built 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 (in the manner of most current web search engines).

Traditional approaches to building distributed systems do not scale well to the large, diverse and growing number of information sources. Technologies such as *Softbot* [12], Sage [19], Occam [20], *ARACHNID* [27], *Meta search engines* [14], Web Robot, Spider, Clower [16], *WebWatcher* [18], and *ShopBot* [10] provide very few capabilities for locating, combining, processing, organizing, and abstracting information about a specific knowledge domain. For these systems several open problems remain:

- <u>Gathering pieces of knowledge</u>: In all of conventional systems, the gathered information is only a set of pieces of knowledge. Science and engineering researchers, as well as decision makers, need more systematic and deeply integrated knowledge on a target domain. For instance, decision makers need a system that provides them with real-time developed knowledge, which includes all the necessary components for decision making. For example, if a decision is to be made for an offensive move toward Baghdad, the system should provide the commanders with comprehensive knowledge. The components of the provided knowledge by the system for such a query could include weather, logistics, maps, pictures, positions (enemy/friendly/neutral troops), etc. Moreover, other related information that may help with the decision making process such as "how operational is the enemy's central command system," should be available. Therefore, a simple one line query to the system should gather, process and categorize the related information for the user.

0-7695-2268-8/05/$20.00 (C) 2005 IEEE

- Client-Server Approach: All the above technologies use traditional client-server approach. In this approach the client computer needs to maintain a continuous communication link with the server. This not only causes both computers (the client and the server) to stay busy with their respective processes until the end of the task, but also they have to continually send and receive messages required for the accomplishment of their task. This imposes a considerable amount of traffic on the network. The client-server model has confronted several challenges, such as problematic legacy network, scaling and protocol problems.

Recently, there is a growing interest in using intelligent agent approach for designing systems that assist users on the WWW. Because of the flexible and dynamic characteristics of intelligent agents, they are being used widely as interface system between the user and the WWW for different applications. For example, Bollacker developed an agent that assists the user on scientific literature search [4], Ackerman an agent that finds web pages for the user [1], and Lieberman an agent for helping users to browse the WWW [22]. Other attempts such as [9] have used a multi-agent approach to help users with common interest to share Web pages, or [25] and [15] which have proposed an agent-based brokering facilitation between users and various information resources. More recently, work by Lesser and associates on agent-based information gathering resulted in the BIG (resource-Bounded Information Gathering) agent architecture [23]. BIG integrates a number of AI technologies, including a real-time planner and scheduler, a task modeling tool, and an information extraction/understanding component (e.g., [6], [24], [13]). The resulting system can reason about resource trade-offs for alternative methods for gathering information, and can potentially use the extracted information to refine its further search and processing activities.

The single-agent approaches are design to assist users for a specific task. These systems are limited to a particular job and are not scalable enough to be expanded to a general information gathering system (and definitely not an information integration one). On the other hand, today's most advanced multi-agent approaches aim for information gathering in the form of pieces of knowledge. These architectures do not systematically offer developed knowledge on a target domain that includes all the necessary components to fulfill a query. They lack seriously the capability of categorizing available information and providing mechanisms to deal with different data formats on the WWW.

In this paper, an intelligent agent approach to design and implementation issues is proposed. We are proposing an integrated system that provides access to a large number of information sources, in a particular domain, by 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. Every information agent contains an *ontology* of its domain of expertise, its *domain model*, and models of the other agents that can provide relevant information, its *information source models*. Similar to the way current information sources are independently constructed, information agents can be 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 many information sources available. An existing data repository can be turned into a simple information agent by building the appropriate interface code, called a wrapper, that will allow it to conform to the conventions of the organization. The advantage of using wrappers is that it greatly simplifies the individual agents since they only need to handle one underlying format.

The rest of the paper is organized as follow. In section 2, the use of intelligent agent as the basic building block of the system is justified. In sections 3 and 4 the architecture of the system is presented and discussed. Finally, section 5 concludes the paper.

2. The Basic Building Block – Intelligent Agents

In order to effectively use the many heterogeneous information sources available in large computer networks, such as the Internet, we need some form of organization. The concept of an agent that provides expertise on a specific topic, by drawing on relevant information from a variety of sources, offers the basic building block.

With the availability of low cost mobile device, such as mobile PCs and PDAs, people are able to access information available on the fixed network anywhere, anytime (providing sufficient transmitter coverage). This means that for any information providing tool to be operational, it should be capable of supporting mobile devices. In the wireless network environments, the mobile devices face several limitations, such as low bandwidth, low computing power, small memory capacity, low battery life, etc., restricting the mobile computing [30]. The advent of the mobile agent technology is expected to overcome these limitations [30]. Mobile agents are specialized independent programs executing on behalf of users. They are transported to multiple remote hosts in the network to carry out assigned tasks. Therefore, potentially they can reduce the communication traffic in the network [30]. This makes them attractive for mobile communications and scalable systems.

Moreover, intelligent agents are differentiated from other applications by their added dimensions of autonomy, and the ability to interact independent of its user's presence. These characteristics make intelligent agents an attractive choice to be used in our model for real-time knowledge discovery and integration [30].

3. General Approach – Network of Information Agents

We believe that a promising approach to distributed information integration is to access the large number of information sources by organizing them into a network of *information agents* [19]. 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. To build such a network, we need an appropriate architecture for a single information agent that can be instantiated to provide multiple agents. An architectures proposed in [26], with some modifications, has been used in our prototype [31] [36].

Similar to the way current information sources 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 many information sources available. An existing information source is turned into a simple information agent by building the appropriate interface code, called a wrapper, that allows it to conform to the conventions of the organization (describe below). A class of wrapper agents would need to be built for any given type of information source (e.g., imagery, graphics, text, formatted text, video and audio, etc). The advantage of this approach is that it greatly simplifies the individual agents since they only need to handle one underlying format. This makes it possible to scale the network into many agents with access to many different types of information sources.

In this system, some agents answer queries addressed to them, but do not actively originate requests for information to others; we will refer to these as *data repositories* (*wrapper agents*). These agents correspond to different information sources including relational databases, object oriented databases, imagery and graphic sources, vector and raster maps, text and formatted text sources, and audio/video sources. For each one of these categories there is an intelligent agent with adequate expertise for knowledge discovery. In the rest of the paper, we will use the term *data repository* or *wrapper agent* when we want to emphasize such agents, otherwise we will use the term *information agent*. Data repositories in our model are described in the next section.

Figure 1 shows an example network of information agents that will be used to explain different parts of the proposed system. The application domain is logistics planning. In order to perform its task, the Logistic_Planning_Agent, which is an information agent, needs to obtain information on different topics, such as transportation capabilities, weather conditions and geographic data. The other agents must also integrate a number of sources of information that are relevant to their domain of expertise. For example, the Sea_Agent combines assets data from the Naval_Agent (such as ships from different fleets), harbor data from the Harbor_Agent and the Port_Agent (such as storage space or cranes in harbors, channels, etc; information that has been obtained, in turn, from repositories of different geographical areas).
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 ontologies of its information sources necessary for the task that it is designed for. For example, the Transportation_Agent might have a fairly complete integration of the Sea, Land and Air_agents, while the Logistics_Planning_Agent might draw on only some parts of the knowledge of the Weather and Geographic_agents. Second, 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 did not include some particular geopolitical facts required by the Logistics_Planning_Agent, the latter could access directly the Geopolitical_ Information_Agent. However, if much of the information was not represented, an alternative geographic agent would need to be constructed (and linked). Third, the network forms a directed acyclic graph, not a tree, because a particular agent may provide information to several others that focus on different aspects of its expertise (like the Port_ Agent that is accessed by the Geopolitical, Air and Sea agents). Nevertheless, cycles should be avoided; otherwise a query may loop endlessly without finding some agent that can actually answer it.

In summary, the network of information gathering agents provides the basis for a rich "knowledge space" built on top of the basic web "data layer". This knowledge layer is composed of value-added services that process and offer abstracted information and knowledge, rather than returning documents. In spite of the complexity introduced by respecting the autonomy of the agents in the organization, the fact that individual agents can be independently built and maintained makes the system flexible enough to scale to large numbers of information sources and adaptable to the needs of new applications.

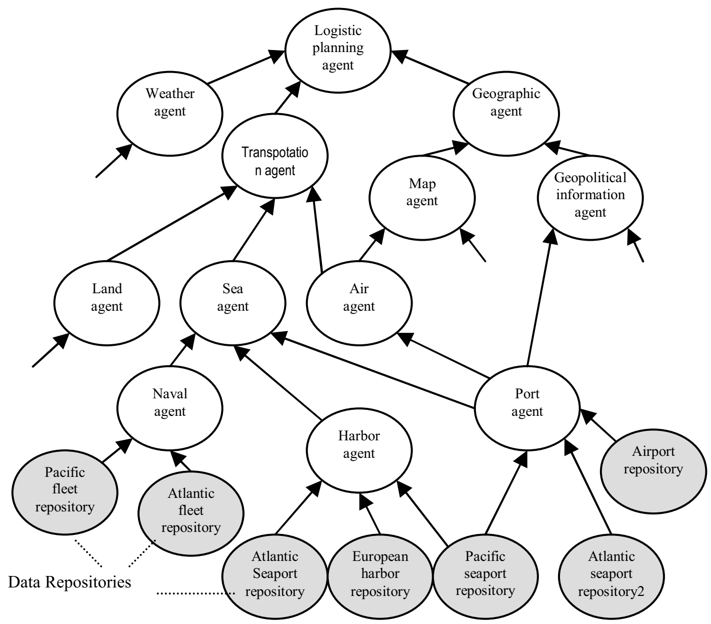

Figure 1. Network of information gathering agents

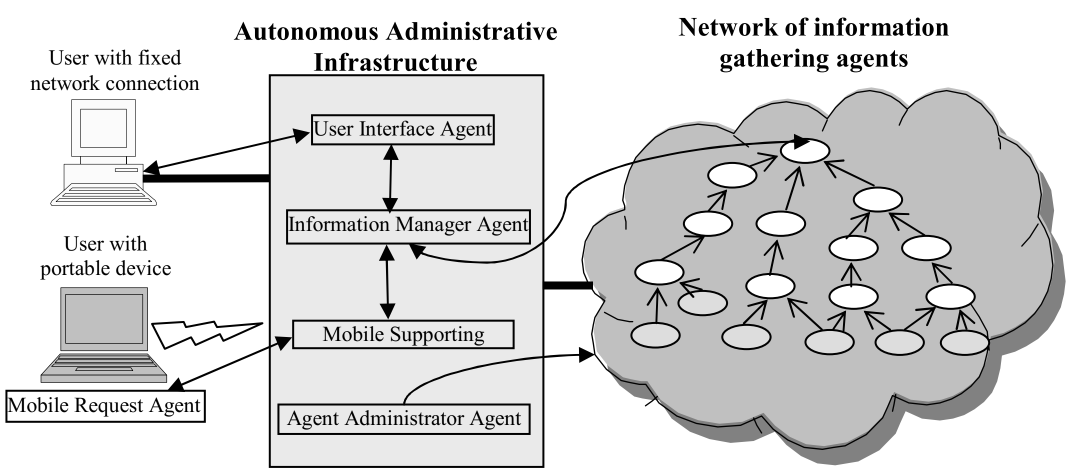

Figure 2. General infrastructure of the complete knowledge discovery system

4. Detailed Approach – The System Architecture

The network of information gathering agents is built 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).

To understand how the autonomous administrative infrastructure works, let's consider a simple scenario (figure 2). A user initiates a query by contacting either a UIA (for users with fixed connection) or a MRA (for users with 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. They provide an interface as the entry point of the mobile network users to the fixed network without a need for high communication traffic. 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 contact 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), constructs the real-time result and sends it back to the IMA. IMA forces some more formatting to the information and send it back to the MSA or UIA to be presented to the user. In case of the portable device users, MSA provides MRA with the findings and launches it 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/modify information agents and therefore their links. The administrator can also introduce new application domains to the system using this agent.

A more complex, but similar, administrative infrastructure for a multi-agent system for geospatial information gathering and integration was successfully designed and developed by the author and his team as part of a five year DoD grant (NIMA-NURI grant NMA202-99-BAA-02) and is currently being used [31].

The main components of the information and wrapper agents are described below.

4.1. 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 an *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 of the agent and establishes the terminology for interacting with the agent. The information-source models describe both the contents of the information sources and their relationship to the domain. These models do not need to contain a complete description of the other agents, but rather only those portions that are directly relevant.

The domain model of an agent defines its area of expertise and the terminology for communicating with it. That is, it provides an ontology to describe the application domain. The ontology consists of descriptions of the classes of objects in the domain, relationships between these classes, and other domain-specific information. These classes and relationships do not necessarily correspond directly to the objects described in any particular information source. The 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 source. This comprises the concepts of interest available from that information source in terms of the ontology of that information source. The terms in the ontology provide the language that the information source understands. Second, the relationship between these information source concepts and the concepts in the domain model needs to be stated. These mappings are used for transforming a domain model query into a set of queries to the appropriate information sources.

These knowledge components were first modeled in API-Calculus [32] and then verified and evaluated using ACVisualizer tool, a validation/verification/evaluation tool for API-Calculus [2]. After finalizing the models, they were stated in XML Declarative Description, XDD in short [35]. XDD employs XML as its bare syntax and enhances XML expressions power by employing Declarative Description theory [3]. A description in XDD is a set of ordinary XML elements, extended XML elements with variables, and the XML elements' relationships in terms of XML clauses. An ordinary XML element denotes a semantic unit and is a surrogate of an information item in the real knowledge domain. An extended XML element responds implicit information or a set of semantic units. Clauses express rules, conditional relationships, integrity constraints, and ontological axioms. XDD is integrated with Java Expert System Shell [17] to facilitate dynamic representation of domain models and information source models in the system.

In our prototype system, "geographical and spatial aspects of military logistic planning" is the knowledge domain of choice for which the network of the intelligent agents is being created. Dr. Fredrick Petry, a professor at Tulane University, who also is a scientist working for Naval Research Laboratory (NRL) at Stennis Space Center and is exceptionally familiar with the above knowledge domain, is the domain scientist of this project [28] [7] [8] [21].

4.2. Query processing

A critical capability of an information agent is the ability to flexibly and efficiently retrieve and process data. Query processing requires developing a plan for obtaining the requested data. This includes selecting the information sources to provide the data, the processing operations, the sites where the operations will be performed, and the order in which to perform them. Since data can be moved around between different sites, processed at different locations, and the operations can be performed in a variety of orders, the space of possible plans is quite large.

For the prototype system, we have developed the basis for a flexible planning system to generate and execute query access plans. Some desirable features of the query processor are the ability to execute operations in parallel, to augment and replan queries that fail while executing other queries, and, most interestingly, to gather additional information at runtime to aid the query processing.

The query processing mechanism is being implemented around our previous query processing unit developed for geospatial information gathering and integration (NMA202-99-BAA-02) [31] which was also used in SWORD, a comprehensive distributed web access juvenile delinquency database (Department of Public Safety-JAIBG Grant).

4.3. Wrapper Agents (Data Repositories)

As was described earlier, wrapper agents are information agents that correspond to different data repositories. For each category of data repositories (information source), there will be an intelligent agent with adequate expertise for knowledge discovery.

Wrapper Agents are responsible for the system's lowest level knowledge extraction (data mining). These agents are the only Information Agents that are required to do actual data mining and knowledge discovery over raw data. As was mentioned earlier, other information agents build their knowledge from the knowledge produced by the wrappers or other lower level information agents, which provide them with abstracted information. The steps of the knowledge discovery process, which is carried out by the wrapper agents, are as follows [33]:

- Data Selection: Creating a target data set in its domain expertise
- Data handling
  - Cleaning and preprocessing
  - Data reduction and transformation
- Choosing functions of data mining algorithm(s)
- Data mining: search for patterns of interest
- Pattern evaluation and knowledge presentation in appropriate domain model
  - Transformation, removing redundant patterns, etc.
- Providing discovered knowledge to higher level information agents

Data mining is the core of the knowledge discovery process and includes the following subtasks [33]:

- Cluster analysis: Cluster engine is used for the automated detecting clusters of records that lie close to each other in a certain sense in the space of all variables. Such clusters may represent different target groups in different domains. The cluster engine places records corresponding to different clusters in separate datasets for further analysis.
- Classification
  - Finding models (functions) that describe and distinguish classes of information
  - Presentation using decision-tree and classification rule
- Association: correlation and causality
- Concept description: characterization and discrimination
  - Generalize, summarize, and contrast data characteristics
- Outlier analysis
  - Outlier is a data object that does not comply with the general behavior of the data
  - It can be considered as noise or exception but is quite useful in rare event analysis
- Trend and evolution analysis
  - Deviation and regression analysis
  - Sequential pattern mining, periodicity analysis
  - Similarity-based analysis
- Other pattern-directed or statistical analysis

Since the above process may generate many patterns - not all of them are interesting to the system - a final step of *interestingness measurement* is also performed. In the architecture of the wrapper agents, a pattern is interesting if it is understandable by the system based on the domain models, valid with some degree of certainty, potentially useful to the system, novel, or validates some hypothesis that the higher level Information agents seek to confirm. The interestingness measurement in wrapper agents is an objective task, based on statistics and structure of the patterns, e.g., support, confidence, etc.

Prototype wrapper agents for vector and raster images have been implemented for our domain of choice (geographical and spatial aspects of military logistic planning). The author and other collaborators in this project have mostly used their previous works in knowledge discovery which were used for the wrapper agents of the geospatial information gathering and integration system, mentioned earlier [31] [36].

4.4. Information Processing in Information Agents

As figure 1 illustrates, the knowledge discovered by the wrapper agents is provided to the higher level information agents in the network for integration and abstraction. Each information agent processes the information it receives from other agents and provides abstract knowledge, consistent with its domain model, which in turn is provided to other agents that request its expertise. This information processing includes the following tasks:

- Data cleaning: smoothes noisy data, removes outliers from its domain model.
- Data integration: Combines data from multiple sources while detects and resolves data conflictions.
- Data transformation: Data normalization and aggregation by using domain model described earlier.
- Data reduction (abstraction): Obtains reduced representation in volume but produces the same or similar results in its domain model.

These information processing tasks are very similar to the ones implemented for the Conflation agents in [31], with this difference that there we were dealing only with geospatial data. The author and his team have used this previous experience for the design and implementation of the information processing unit of the information agents of the prototype system.

4.5. Control and Coordination

Ultimately, control and coordination will be key issues in multi-agent systems for real-time information discovery and integration. Real-time constraints generally require that problem solving be approximate and satisfying, where certain aspects of the quality of the results (e.g., breadth or precision) must be traded-off to meet the imposed time constraints. In addition to being a multi-attribute decision problem, control here also involves considerable uncertainty. The quality characteristics of results for any query can be predicted only approximately, and the response times and even availability of information sources can be highly variable and dynamic. This can require that sources and methods be adjusted in real-time, or that additional information sources be exploited if results prove unsatisfactory. Making such control decisions can be extremely complex, since the behaviors of multiple agents will need to be coordinated.

For example, should one agent/source prove to be unacceptably slow in providing results for a detailed subquery, it may be desirable to modify that agent's subquery. This will also require that other agents' subqueries be modified as is appropriate to best meet the overall system goals. As discussed in query processing section, query planning largely ignores these issues initially, by assuming that providing agents can make good predictions, in their domains, about performance, etc. However, the question of what approaches to control and coordination are most appropriate for multi-agent real-time information discovery and integration will be explored in parallel. In our prototype network, we have not considered this issue, but for the next versions, we will consider both heuristic approaches such as real-time planners [23] and [5], as well as formal methods such as decentralized Markov decision problems [34].

5. Conclusion

Information sources are becoming more diverse and more technically capable. Because of fast advancing information technology, scientists, educators, and decision makers are facing much more complex and detailed questions. To answer these questions, they need to be multidisciplinary experts or have access to experts in many specialized disciplines. The proposed system models the above relationship by using intelligent agents and distributing the knowledge discovery process on a particular domain among multiple agents responsible for different sub-domains.

A prototype system based on the proposed architecture has been implemented. In our prototype, "geographical and spatial aspects of military logistic planning" is the knowledge domain of choice for which a preliminary network of three wrapper and five information agents has been created (at SIUC Software Agent Lab). The preliminary evaluation of the system looks promising and we are planning to publish these results shortly.

6. Reference

[1] Ackerman, M. S., Starr, B., and Pazzani, M. (1997). The Do-I-Care Agent: Effective Social Discovery and Filtering on the Web. *Proceedings of RIAO Computer-Assisted Information Searching on the Internet*, pp. 17-31.

[2] Ahmed, R., Ali, D., Cobb, M., and Rahimi, S. (2003). A Visualization Tool for the Intelligent Agent Modeling Language API-Calculus. *Proceedings of the 9th Annual International Conference on Industry, Engineering, and Management Systems*, pp. 482-488.

[3] Akama, K. (1993). Declarative Semantics of Logic Programs on Parameterized Representation Systems. *Advances in Software Science and Technology*, Iwanami Shoten Publications and Academic Press, Tokyo, vol. 5, pp. 45-63.

[4] Bollacker, K., Lawrence, S. and Giles, L. (1998). CiteSeer: An autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications. *Proceedings of the 2nd International ACM Conference on Autonomous Agents*, pp. 116-123.

[5] Carver, N. and Lesser, V. (1993). 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.

[6] Carver, N. and Lesser, V. (2003). Domain Monotonicity and the Performance of Local Solutions Strategies for CDPS-based Distributed Sensor Interpretation. *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 6, # 1, pp. 35-76.

[7] Chung, M., Wilson, R., Cobb, M., Petry, F. and Shaw, K. (2001). Querying multiple data sources via an object-oriented spatial query interface and framework. *Journal of Visual Languages and Computing,* vol. 12, #1, pp 37-60.

[8] Cobb, M., Petry, F., Wen, L. and Yang, H. (2003). Design of system for managing fuzzy relationships for integration of spatial data. *Fuzzy Sets and Systems*, vol. 140, # 1, pp. 51-73.

[9] Davis, J., Week, R. and Revett, C. (1999). Information Agents for World Wide Web. *BT Technology Journal*, vol. 14, #4, pp. 105-114.

[10] Doorenbos, R. B., Etzioni, O. and Weld, D. S. (1997). A scalable comparison-shopping agent for the World-Wide Web. *Proceedings of the First International Conference on Autonomous Agent*, pp. 39–48.

[11] Eirinaki, M. and Vazirgianis, M. (2003). Web Mining for Web Personalization. *ACM Transactions on Internet Tehnology (TOIT)*, vol. 3, #1, pp. 1-27.

[12] Etzioni, O. and Weld, D. (1994). A SoftBot-based interface to the Internet. *Communication of the ACM*, vol 37, #7, pp. 72–76.

[13] Fisher, D., Soderland, S., McCarthy, J., Feng, F., and Lehnert, W. (1996). Description of the UMass Systems as Used for MUC-6. *Proceedings of the 6th Message Understanding Conference*.

[14] Hu, C. L. and Chen W. E. (1998), Mobile Agents Collaboration for Information Gathering. *Workshop on Distributed System Technologies & Applications*, NCKU, R.O.C, pp. 537-546.

[15] Jamsa, K., Lalani, S. and Weakley, S. (1996). *Web Programming*. Jamsa Press.

[16] Jess Manual 6.1 (2003). The Java Expert System Shell. *URL: http://herzberg.ca.sandia.gov /jess*.

[17] Joachims, T., Freitag, D. and Mitchell, T. (1997). Webwatcher: A tour guide for the world wide web. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pp. 770–775.

[18] Kenoblock, C. A. (1995). Planning, executing, sensing, and replanning for information gathering. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 1686–1693.

[19] Kenoblock, C.A. and Ambite, J.L. (1997). Agents for Information Gathering. *Software Agents*, J. Bradshaw, ed., MIT Press, Menlo Park, Calif.

[20] Kwok, C. T. and Weld, D. S. (1996). Planning to gather information. *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 32–39.

[21] Ladner, R., Petry, F., Cobb, M. (2003). Fuzzy set approaches to spatial data mining of association rules. *Transactions on GIS*, vol. 7, #1, pp. 123-138.

[22] Leberman. H. (1995). Letizia: A agent that assists Web browsing. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 924–929.

[23] Lesser, V., Horling, B., Klassner, F., Raja, A., Wagner, T., and Zhang, S. (2000). BIG: An Agent for Resource-Bounded Information Gathering and Decision Making. *Artificial Intelligence Journal*, Special Issue on Internet Information Agents, vol. 118, # 1-2, pp. 197-244.

[24] Lesser, V., Decker, K., Wagner, T., Carver, N., Garvey, A., Horling, B., Neiman, D., Podorozhny, R., Prasad, M., Raja, A., Vincent, R., Xuan, P., and Zhang, XQ. (2004). Evolution of the GPGP/TAEMS Domain-Independent Coordination Framework. To appear in *Journal of Autonomous Agents and Multi-Agent Systems*.

[25] Martin, D., Morgan, D., Gohama, H. and Cheyer, A. (1997). Information Brokering in an Agent Architecture. *Proceeding of the second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, pp. 467-486.

[26] Matsatsinis, N.F., Moratis, P., Psomatakis, V., Spanoudakis, N. (1999). An Intelligent Software Agent Framework for Decision Support Systems Development. *Proceedings of European Symposium on Intelligent Techniques*. Greece, pp 134-139.

[27] Menczer, F. (1997). ARACHNID: Adaptive retrieval agents choosing heuristic neighborhoods for information discovery. *Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 227–235.

[28] Murray, B. H., (2000). Sizing the Internet, A Cyveillance White Paper. *URL: http://www.cyveillance.com/web/downloads /Sizing_the_Internet.pdf.*

[29] Petry, F., Cobb, M., Rahimi, S., Ali, D., Paprzycki, M., and Angryk, R. (2002). Fuzzy Spatial Relationships and Mobile Agent Technology in Geospatial Information Systems. *Soft Computing in Defining Spatial Relations*, volume in series: Soft Computing, Edited by Pascal Matsakis, and Les M. Sztandera, Phsica-Verlag, pp. 123-155.

[30] Rahimi, A., Ali, A. and Ali, D. (2001). An Investigation on Intelligent Software-Agent Technology. *Proceeding for IEMS and IC&IE 2001 joint international conference*, Florida, pp. 84-90.

[31] Rahimi, S., Cobb, M., Ali, D., Paprzycki, M.and Petry, F. (2002b). A Knowledge-Based Multi-Agent System for Geospatial Data Conflation. *Journal of Geographic Information and Decision Analysis*, vol. 6, # 2, p.p 67-81.

[32] Rahimi, S., Cobb, M., Ali, D. and Petry, F. (2002c). A Modeling Tool for Intelligent-Agent Based Systems: Api-Calculus. *Soft Computing Agents: A New Perspective for Dynamic Systems,* the International Series "Frontiers in Artificial Intelligence and Application," Edited by Vincenzo Loia, IOS Press, pp. 165-186.

[33] Rahimi, S. (2003). Data Mining: Concepts and Technology. Technical Report, Southern Illinois University, Carbondale.

[34] Shen, J., Lesser, V., Carver, N. (2003). Minimizing Communication Cost in a Distributed Bayesian Network using a Decentralized MDP. *Proceedings of Second International Joint Conference on Autonomous Agents and MultiAgent Systems*, pp. 678-685.

[35] Wuwongse, W., Anutariya, C., Akama, K., and Nantajeewarawat, E. (2001). XML Declarative Description (XDD): A Language for Semantic Web. *IEEE Intelligent Systems*, vol. 16, # 3, pp. 54-65.

[36] Zhou, H, Rahimi, S., Wang, Y. and Cobb, M. (2004). A Task-Oriented Compositional Mobile Agent Architecture for Knowledge Exchange between Agencies and Agent. *Informatica Jurnal, Special Issue on Agent-Based Computing,* pp. 23-30, Vol. 28, No.1.