# Intelligent Information Agents:

# Review and Challenges for Distributed Information Sources


Donna S. Haverkamp

Susan Gauch

Department of Electrical Engineering and Computer Science

The University of Kansas

{haverkam,sgauch}@eecs.ukans.edu

Contact Information:
Dr. Susan Gauch
Department of Electrical Engineering and Computer Science
415 Snow Hall
Lawrence, KS 66045
913-864-8817
913-864-3229 (fax)
sgauch@eecs.ukans.edu

**ABSTRACT**

Inexperienced users typically obtain one of three possible outcomes when they search for online information: they are buried under an information avalanche, they are unable to locate *any* useful information at all, or they find what they need in roughly the amount they need. Unfortunately, the latter outcome is the most rare. Unfamiliarity with search tactics creates difficulties for many users of online retrieval systems. When faced with poor results, even experienced searchers may use vocabulary incorrectly and often fail to reformulate their queries. Far from being the answer to everyone's information dreams, distributed sources of online information, i.e., the World Wide Web (WWW), compound the problem and may often turn into an information nightmare. To address this problem, intelligent online search assistants, or *agents*, are being developed for information retrieval applications. There are many approaches, both theoretical and implemented, to using intelligent software agents for information retrieval purposes. These approaches range from desktop agents specialized for a single user to networks of agents used to collect data from distributed information sources, including Web sites. This paper presents an overview of intelligent software agents in information retrieval, including an explanation of agents and agent architectures, and presents several agent systems. We distinguish between *agents* as individual entities, whose properties and characteristics we describe separately, and *agent systems* as collections of agents utilized for information retrieval tasks, which we discuss in terms of individual implementations.

## 1 INTRODUCTION

Technology to produce, store, and distribute massive quantities of electronic information has matured. The Information Highway is becoming a reality. The increase in access to the Internet by the public at large, combined with the development of easy to use graphical browsing interfaces (for example, Mosaic and Netscape) has led to an explosion in the amount of information being added. In particular, the World Wide Web (WWW) provides an exponentially growing amount and range of information through which people can browse. A few dozen bibliographic databases have mushroomed into several thousand databases of everything from full-text documents to movie clips. To give a feel for the magnitude of the growth, consider that 100,000 artifacts were Web-accessible in 1994. By 1995, 1.5 million artifacts were Web-accessible[1], and today there are over 30,000,000 Web pages on 225,000 Web servers[2]. Demand for this information has also exploded. For example, the WWW Worm search system received 2,000,000 search requests per month in 1994[3]. However, the Alta Vista search system now receives 10,000,000 search requests *daily*[2].

In the early days of online information retrieval systems, individuals met with search intermediaries who were trained to use the online systems and were often knowledgeable about the information seeker's area of interest. Through an interview process, the search intermediary would determine the individual's information need, perform the actual searches, and send the results to the information seeker. Technology, in the form of personal computers and networks, now provides many people with the means to access the online databases from their own offices. However, both Borgman and Fenichel have found that merely giving users the ability to search is not enough; even experienced users must also be given assistance with the search process itself[4,5].

Technology influences the amount and type of information available, but it must also provide the means to make effective use of this information from users' homes and desks. The research community could make a significant contribution by developing systems which allow end-users to search effectively. That is the goal of intelligent search agents, whether they search a single database of bibliographic records or a network of distributed, heterogeneous, hypertext documents.

We begin with a discussion of intelligent user interfaces, through which a user may query for information. We continue with a general discussion of agents to familiarize the reader with basic agent structure and communication techniques. We then have an explanation of the different types of *agents* which are used to support intelligent user interfaces. Different types of *agent systems* can be identified using a simple taxonomy which we have developed in order to characterize them. Specific information agent system examples from literature are selected as representative of different possible agent combinations within that taxonomy. We conclude by discussing the current state of agents in information retrieval, especially as influenced by the World Wide Web, and possible future directions.

## 2 INTELLIGENT USER INTERFACES

In an earlier paper by Gauch[6], research focusing on the development of intelligent front ends to traditional information retrieval systems is summarized. In general, these systems attempt to provide *conceptual retrieval*, rather than simple string matching. The user's initial query terms are taken as representatives of the concepts in which the user is interested. Query expansion then adds other terms related to the same concepts, providing a richer representation of the user's query. For example, a user query of "dog" can be expanded to "dog canine beagle" given an appropriate knowledge base of related words.

Expert systems have been developed which assist the user by automatically expanding the terms in the user's queries from an online thesaurus or knowledge base which stores word relationships. These expert systems incorporate the search strategies of expert human searchers. They may be customized for particular domains such as cancer treatment[7] or environmental pollution[8], or they may be based upon domain-independent search strategies[9]. In our earlier work, the thesaurus of query terms is hand-built[10], a time-consuming and ad hoc process. In other cases, it is an online version of a published thesaurus or semantically coded dictionary[11]. As exhibited by Gauch & Wang[12] and others[13,14], current research focuses on constructing the knowledge base of related words automatically.

## 3 INTELLIGENT SEARCH AGENTS

The expert systems discussed in the previous section were, by and large, designed to work with single, independent databases. With the emergence of the Internet, there are many, many databases stored in many different locations. Obviously, there is no sense in replicating remote databases permanently at each local site due to space requirements and the cost of transporting it all over the Internet, nor do you want to manually search all databases or sift through Web sites for a piece of desired information. The distribution of the information sources leads naturally to the desire for a distributed approach to obtain data from those sources. The distributed problem solving method currently being investigated is the use of intelligent software agents which can locate, retrieve and integrate the answers into one “result” for a given user. Therefore, this section will discuss the following related topics: what search agents are (i.e., what properties, abilities, and characteristics do they possess, how do they communicate), how they are arranged to interface with a database or other information source, how successful they are in retrieving information (exhibited through a series of example systems), and what the future of information retrieval holds with respect to intelligent agents.

### 3.1 Agents and Distributed Problem Solving

Distributed problem solving (or distributed artificial intelligence) is based upon the idea that a goal or problem can be split into subgoals or subproblems, and is concerned with how a problem can be divided among a number of modules (or agents) to achieve a solution[15]. Intelligent entities or *agents* can effect solutions for subproblems using their unique expertise, and these subproblem solutions can then be merged into a complete solution. In the case of information retrieval, this implies the splitting of a query into subqueries. An agent who “knows” something about a particular subquery may take on the task of fulfilling that subquery. For example, if the query is “dogs and cats”, and there is an agent associated with an entire database about dogs, then that agent may offer to take on the subquery of “dogs”. After separate results are obtained for “dogs” and “cats”, they are merged and presented to the user. Using agents, the query is split into subqueries, and the results of all subqueries are re-combined to obtain the information “solution” to the entire query.

In this section, we explain agents in a general sense by discussing the characteristics they possess and their methods of intercommunication, concluding with a discussion of the newest type of agent on the scene, the mobile agent.

Software Agents and Information Agents

Agents themselves may be regarded almost as individual entities - pieces of software that control their own lives. They are (usually) continuously-running processes that know what to do and when to do it. They communicate with other agents, making requests and performing requested tasks. According to the artificial intelligence point of view, an agent possesses a long list of properties[16], including: autonomy; social ability; reactivity; pro-activeness; benevolence (I'm such a nice agent - I always do what someone asks me to do if I can); and rationality (I always behave in a rational manner). The possession of all these properties may not be necessary, however, for an agent used in information retrieval. Benevolence and rationality may be assumed or simply ignored. The information retrieval view of an agent, according to Wooldridge and Jennings, can be limited to the following characteristics[16]:

| | |
|---|---|
| *autonomy* | agents operate without direct human intervention and have control over their own actions |
| *social ability* | agents communicate through a common language to other agents and even to humans |
| *reactivity* | agents perceive their environment and react to changes in it |
| *pro-activeness* | agents are capable of exhibiting goal-directed behavior by taking the initiative |

Given the above information retrieval view of an intelligent agent, we will hereafter refer to these agents not as intelligent agents for information retrieval, but simply as *information agents.*

An information agent possesses the properties of autonomy, social ability, reactivity, and pro-activeness, as stated above. These agents are designed specifically for query processing. An agent is made up of at least some of the following components: processing power, knowledge of environment, domain models, and information models. An agent possesses processing power in order to perform planning (it must often split a query into subqueries) and query processing (it may have to change or interpret queries). Its knowledge concerning the environment pertains to knowledge of itself and other agents - it knows what information it has access to, and it may know what other agents to ask for information about keyword *k*. An agent may also have access to domain models and/or information source models if the agent is associated with a structured domain, particularly relational databases.

## Cooperative Distributed Agents

In addition to being self-reliant and knowledgeable of their environment, agents must somehow coordinate and cooperate with each other in order to achieve a goal (satisfy a query). There are a number of ways in which this may be done[17,18]. In a *specification sharing agent system*, a single agent decomposes the problem and assigns subgoals to other agents. Each agent knows what every other agent's needs and capabilities are. There is no "master" agent - whichever agent receives the query subdivides it among the group of agents. It is assumed that any agent will perform a task requested of it, and they all work together as equals to achieve a solution.

In *contract nets,* all agents are still equal. The query-receiving agent does not, however, assign subqueries to other agents. Instead, the agent decomposes the problem and then *negotiates* with other agents to determine who will perform which subtasks. The agent receiving the query broadcasts his information needs to other agents.  Agents capable of fulfilling subqueries then submit bids to the originating agent (these bids may be a cost function related to necessary Internet links or estimated database access times), and then the originating agent selects agents to fulfill the subqueries based upon the various bids that it receives.

In contrast, a *federated system* is a hierarchical arrangement of agents, where each federation of agents is controlled by a facilitator, or master agent.  The federated agents communicate only with the local facilitator, which knows the needs and capabilities of every agent in its federation.  The facilitators intercommunicate, broadcasting the needs (requests) of their local agents and passing only the relevant requests from other facilitators to the proper local agent.

For any kind of cooperation or coordination to take place, the agents must have a way of communicating their needs and capabilities to one another.  There are two ways of doing this.  In *message-passing systems*, the communication may be direct, as in specification sharing and contract nets, or it may be indirect, as in federated systems (the agents communicate only through the local facilitator).  Direct message-passing has the disadvantage of both high cost and implementation complexity.  For example, in a system with 10,000 agents, an agent who broadcasts a message sends it to 9,999 other agents, creating quite a load on the network.  If those 10,000 agents were divided into 100 federations of 100 agents each, however, that same message would be broadcast to only 999 facilitators.  Assuming that only a few agents in the entire system of 10,000 have the needed capabilities, only a few messages will be passed from those 999 facilitators down to local agents.  The communications load would then be reduced by nearly 90%.

The other method which agents often use to communicate is through *blackboards*. Instead of messages being sent directly from agent to agent, they are posted to a communal blackboard which is monitored by every agent[19]. Both requests for performing tasks (or processing subqueries) and the results of processing are placed on the blackboard for all to see. This may be likened to a roomful of people all working on the same problem, but not allowed to speak to one another. Any time that someone needs a piece of information or has an answer to a piece of the problem, she/he runs up to the blackboard and scribbles it down. The other people in the room look at that request or partial solution and decide whether they can provide the necessary information or use the partial solution to solve another piece of the problem, and they act accordingly in order to achieve a final solution.

### Mobile Agents

One of the newest developments in agent technology is the concept of a mobile agent[20]. A mobile agent can suspend processing, transport itself across the network, and resume execution elsewhere. This unique capability allows such an agent to sift through retrieved data *before* transmitting it over the network, thus avoiding the transfer of large amounts of data which may be ultimately rejected anyway. The fact that it transmits itself across the network and resumes execution at a different location implies several issues particular to a mobile agent. One is that the code used to create a mobile agent must be architecture independent, suggesting the use of an interpreted language such as Java, Perl, or KQML. Another issue is security: how nervous would your system administrator be if agents were popping up and executing who-knows-what on the local machines? One way to prevent the execution of unsafe arguments is to use Safe-Tcl, which distinguishes between a trusted and an untrusted environment. A mobile agent is allowed to run only in the untrusted environment and is limited to only safe calls. Even if an agent does not pose a security risk, the idea of large numbers of agents suddenly executing and stealing execution time from the local users is not an attractive one. It may be necessary to impose limits on the number of mobile agents allowed to execute concurrently on a local system.

### 3.2 Information Agents for Database Information Retrieval

Initially, information agents were developed for searching distributed relational databases. More recently, agents have been developed which search for textual information from Usenet news articles or the World Wide Web. Regardless of the type of information searched, the search agents can be categorized based on whether agents are active on behalf of individual users and/or queries and/or databases. They also differ based on whether or not the agents for the user (or query) interact directly with the database agents, or if there are special agents (mediators) which act as go-betweens.

We identify different types of agent systems by the different agents of which they are composed; or, more specifically, the functions of those agents as just described above. The taxonomy is thus arranged with respect to the agent function utilized: agents acting on behalf of a submitted query vs. a particular person or user, agents acting on behalf of a local database to supply information vs. direct database queries, and agents acting as go-betweens for query agents and database agents vs. direct communication between query and database agents.

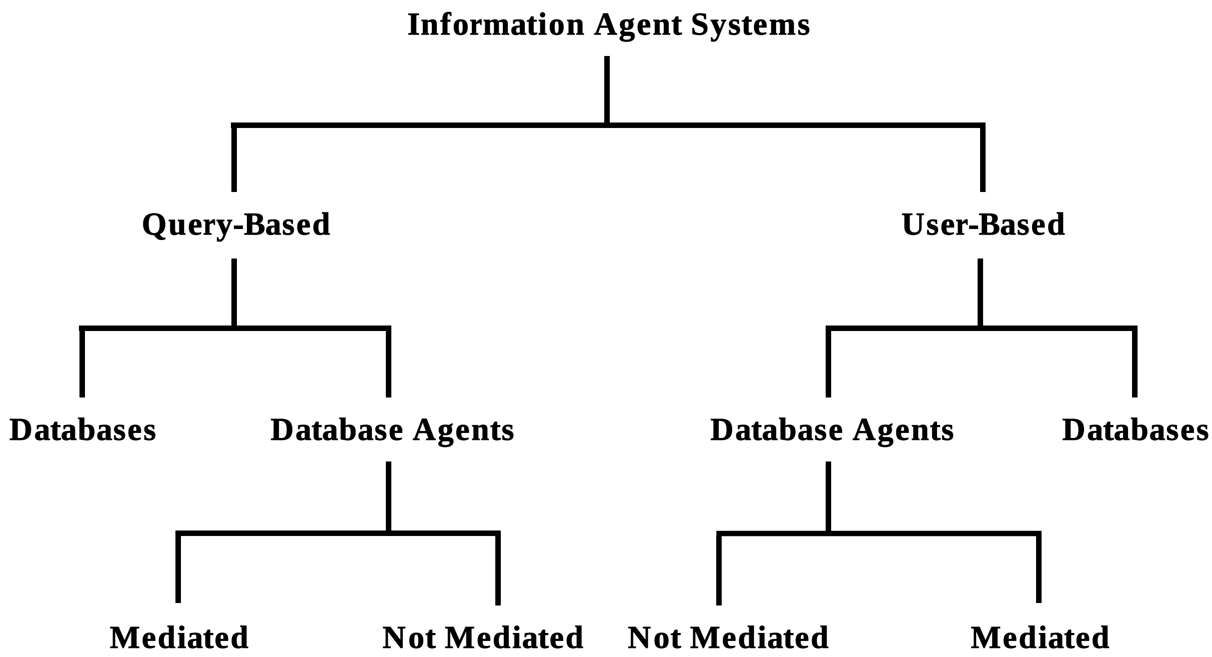


**Figure 1. Information Agent Taxonomy**

Query Agents

Query-oriented agent systems spawn one or more agents in response to user queries. These agents work on behalf of the user for the duration of that query, gathering information from multiple databases. The Cooperative Information Gathering[21] system (or CIG) is one such framework in which each local network has an associated agent which interacts with local information sources and incoming queries. A query to a local network agent results in query submission to local information servers and/or the spawning of agents on other networks. The new agents may in turn submit the query (or pieces of it) to local information server agents or spawn query agents on other networks again. The CIG concept concentrates on distributed information retrieval and methods for achieving such retrieval.

Database Agents

In the above architecture, databases are passive repositories of information. In contrast, however, queries may be passive and the databases themselves may have active agents which interact with individual queries. Each database has an agent which knows how to properly submit queries to its corresponding database and knows what kind of information is stored in the database. Intelligent and Cooperative Information Systems (ICIS)[22] is one such example system which has an agent associated with each database. The database agents are organized into federated agent clusters. The agents within a cluster have knowledge of other agents within the same cluster (*close acquaintances*), but little knowledge of agents outside their own federation (*remote acquaintances*). When the primary agent (or facilitator) of a federated cluster receives a query, it assigns pieces of that query to appropriate members of its own federation. Then, other pieces of the query which cannot be satisfied using local agents are broadcast to remote clusters for a contract net negotiation. Because it knows little about the other federations, the primary agent uses knowledge of global schema to submit the query or subqueries to the other clusters. Thus, the primary agents are required to have planning capabilities in addition to result merging capabilities.

Query and Database Agents

Databases and queries may both have agents which interact in order to retrieve information in response to user requests. A pioneering example of this type of system has been developed for relational databases[23]. An *is-a* hierarchy acts as a knowledge structure which maps from concepts in the domain to the appropriate information sources. Query agents receive user queries (expressed in terms of the domain model) and map from the domain model to the information source model in order to find out which database agents can satisfy that query. Based upon the least expensive path to the goal of query satisfaction, the query agent selects the database agents best suited to provide the needed information. For example, if the query consists of two keywords, and there is an agent that has knowledge about one of the keywords, but there is a second agent that has knowledge about *both* keywords, then the second agent is preferred because one query submission is less expensive than two. This implies that the query agents have planning capabilities which allow them to select the least expensive course of action. Once a query has been submitted to the proper database agent, that agent may alter the query so that it has the proper syntax for communicating with and retrieving information from the corresponding information source (database). This system also has learning capabilities and is operational on a set of nine relational databases.

### Mediated Query and Database Agents

Mediator systems extend the query / database agent model above to include a new type of agent, the *mediator* agent, which mediates between the query agents and the database agents. One recent project is MACRON (Multi-agent Architecture for Cooperative Retrieval ONline)[24]. Its agents use information sources available on the Internet such as newsgroups, archives, magazine databases, and even corporate WWW sites. It has an organizational architecture and uses reasoning agents, low-level network retrieval agents, and user interface agents. The architecture consists of functional and query-answering units, each of which are made up of a number of individual agents, including a facilitator. Functional units provide access to information sources (like newsgroups, archives, etc.), while query-answering units consist of a query-manager agent and a set of agents selected from the set of functional units to process a given user’s query. It is the query-manager agent which creates the information-gathering plan and the functional unit facilitators which supply the functional agents for performing each subtask.

User (Desktop) Agents

User-oriented agents are those associated with a particular person. In contrast with query agents which are spawned in response to user queries, user agents are always active, searching information out and bringing it back to their creator. Query agents are applicable to information retrieval tasks wherein the information sources are relatively static but the information desired is dynamic. Conversely, user agents are more suited to information filtering tasks wherein the information needs are relatively constant but the information sources are dynamic. User agents can be used to retrieve documents from databases, to pick potentially interesting items off the electronic news, or even to sift through e-mail. Their purpose is to decrease the work necessary for their user to retrieve information; in effect, to make office life easier. The personal agent may learn from user feedback through direct positive or negative reinforcement: when it presents its user with retrieved documents, the user may select certain ones and indicate that she/he is or is not pleased with the retrieval of those documents. Learning may also be achieved through genetic algorithms[25], in which the agent is essentially a population of profiles. Based on user response, the fitness of a profile is increased or decreased, and then crossovers and/or mutations are performed on the most fit to create a new population.

User and Database Agents

This type of system is typically characterized by a combination of personal user agents and information source (database) agents. The personal user agents interface between the users and the database agents to retrieve information for the system users. One early and rather unique example of this type of system is MINDS

(Multiple Intelligent Node Document Servers)[26]. Each networked workstation in an office environment has a collection of documents which may migrate to other workstations in the office. Each of these workstations also has an associated agent to interface to the collection of documents; and because it is assumed that each workstation has one primary user, each workstation agent also doubles as a user-agent for the local user. The agent has a local user model which represents which documents the local user likes, and it associates document keywords with other agents in the networked office system. In this way, it has both a user model and a *belief system* pertaining to other agents. For example, if Agent 132 requests information about John Doe from Agent 22, there is one of two outcomes: if Agent 22 returns good information, then Agent 132's belief in Agent 22's ability to produce information about John Doe is increased; on the other hand, if Agent 22 returns bad information, then Agent 132's belief in Agent 22's ability to produce information about John Doe is decreased. Communication is achieved by maintaining query and response blackboards at each workstation. Whenever an agent wishes to make a request of another agent, it posts the request on its local query blackboard; conversely, when it has the results of a query, it posts it on its local response blackboard. The network server delivers messages among the agents, picking them up from originating blackboards and depositing them on the proper target blackboards.

InfoScout[27], another example, is an information management system which is implemented as a set of agents which represent users and data collections. The developed prototype performs text retrieval in large networks in which each user has a *userbot* and each text collection is represented through a *corpusbot*. Its goal is to provide users with a personalized retrieval tool and access to the expertise of other users' search methods. It allows users to benefit from the experience of other users by providing a means to examine how other users accomplish their successful information searches. Each userbot is a profile which contains information for its user which can be examined and imported by other users. It has knowledge of the areas of expertise and interest of its user in addition to other information, but its autonomous capabilities are little to none. If its user wants to locate someone with expertise in a certain field, the userbot can broadcast queries to find other users who are experts in that area. Then the originating user can submit queries to the discovered expert through their respective userbots - the information that is returned can indicate which corpusbot to query and/or how to properly formulate such a query in order to retrieve the best information. Users can customize and extend the functionality of their userbots through the use of scripts which give the userbots instructions for often-repeated tasks. The corpusbots simply accept queries and invoke a retrieval engine to search their associated databases, returning the results through the communication language.

Mediated User and Database Agents

Matchmaker systems implement a world in which a mediator agent acts as the go-between for information consumers (users) and information providers (sources)[28]. The mediator agent functions as a facilitator - it knows the needs of its participating information consumers and the capabilities of its participating information providers. The mediator may recommend a provider for a consumer, then let the queries take place between the provider and consumer without further intervention. Alternatively, the mediator may act as a recruiter, forwarding a request from a consumer to a provider with the stipulation that replies will be sent directly to the consumer. Yet another possibility is that the mediator acts as a broker, forwarding both requests and replies so that no direct communication takes place between information consumer and information provider. SHADE and COINS are two matchmaker systems which have been utilized in a number of engineering-related projects at various research and industry laboratories, including DARPA and Lockheed[28].

### 3.3 Intelligent Search Agents for the World Wide Web

One important category of search agents are those specifically designed to locate information available from various home pages on the World Wide Web. These agents mask the complexity of the Information Superhighway and filter the exploding amount of information available[29]. The search may be for text, it may be for product information or travel information, or for any number of categories which are represented on the Web today. Research in search agents for use on the Web is still in its infancy. Following are a series of example systems of World Wide Web search agents. They range from general interfaces for Web interaction to online shopping agents.

Web Interfaces

Web interfaces create a buffer between user and Web to alleviate the necessity of performing low-level searching tasks. They allow a user to make a high-level request which is then performed by the Web interface agent. One Web search agent is the Internet Softbot[29]. It differs from previous systems for assisted browsing and information retrieval on the Web in that it accepts high-level user goals and decides on the sequence of actions (Internet commands) required to satisfy those goals. It deals with structured information services such as weather map servers and stock quote servers, and it can handle notification requests by monitoring any of a variety of events and reporting important occurrences to the user *autonomously*. It is capable of acting upon the world around it by compressing files or changing protections on groups of files; it can also compile code, convert documents to different formats, and access remote databases to obtain information.

The Softbot consists of a task manager which resembles an operating system scheduler, a planner which can perform planning with incomplete information, a model manager that stores everything Softbot knows about the world, and an Internet domain model which contains declarative encodings of actions available to the Softbot via the Internet. The Softbot is highly autonomous, reactive, and pro-active. It typically survives for only a few days before rebooting is necessary, however. The makers of the Softbot forecast that, eventually, the World Wide Web will become like an invisible entity which will be accessed only indirectly through agents.

Web Shoppers

The number of online shopping facilities grow as the World Wide Web grows. It would take a great deal of personal time to visit various online stores one by one to find the best price for a single product. If a group of agents could perform this task, the result would be both a time savings and, possibly, monetary savings for the shopper. ShopBot is a comparison-shopping agent which performs comparison shopping at online Web product vendors[30]. It uses a combination of heuristic search, pattern matching, and inductive learning techniques to extract information from online vendors; then it uses the vendor information it has learned to comparison shop for a user. It is functional for shopping at online software shops and CD stores.

In the learning phase, ShopBot analyzes online vendor sites to obtain a symbolic description of each site. This is done one time for each store, which suggests potential future problems if a vendor should suddenly change its page format.

Supplied with a set of URLs for the home pages of possible vendors and a domain description including product attributes which can be used to discriminate among different products and among variants of the same product (like the name of the product, the manufacturer, the price, etc.), it finds and determines how to fill in a search form for any vendor and to pick out the product descriptions from the page. It uses heuristics to determine forms which are definitely *not* searchable indices for product information. Then, using descriptions of popular products in the domain, it forms test queries using more heuristics in order to learn how to fill in the search forms. Finally, product information is identified by analyzing the search form results according to individual lines - the line with the highest ranking (based upon text content, presence of price information, and presence of attributes of the popular test product) is selected as the line which must contain the product description. A vendor description is then created from the learned information; this description includes the URL of the page containing the searchable index, a mapping which associates product attributes to the fields of the search form, and functions for extracting product information from the pages returned by the search form. ShopBot works directly with HTML. If vendor information is embedded in graphics or if it uses Java, ShopBot cannot work with that vendor. Also, its scope of vendors is limited to those that provide a searchable index for their products. It is intended that these drawbacks will be overcome in the future.

After learning about each vendor, ShopBot enters the comparison shopping phase, in which it uses the learned vendor descriptions to shop at each site and find the best price for a specific product requested by the user. Given the product name and manufacturer and an attribute X (such as price) by which to compare the vendors, it determines the set of stores where the desired product is available, sorted by X. An example request would be: “Find the lowest price for the latest PC version of

Microsoft Word." It would then perform a parallel search on all possible vendors to obtain the product information and return it in a list ranked according to price. It greatly reduces the amount of work (and time) required by a human online comparison shopper.

Anticipated Web Agent Capabilities

Some researchers project a future in which people will have very little direct interaction with the World Wide Web[29]. Instead, they will have agents to take care of all their searching and arranging of schedules through the network. This would require a system of agents which can deal with interdependencies among the subproblems assigned to each of them[31]. An example of such a system would be a multi-agent based travel agent system. A user would call upon this travel agent to investigate possible transportation arrangements, hotel reservations, and rental car reservations. This travel planner might also know what sort of surroundings the user prefers (nature spots, historical sites) from either user modeling or user specifications. An agent would be assigned to each of the aspects of a potential vacation or business trip, and an additional agent would be used to check on weather information (if the weather is going to be bad, it may be preferable to take a bus or train instead of an airplane). These agents would intercommunicate their information (or constraints) in order to concoct an overall plan for the trip.

## 4 CONCLUSIONS

The number of information sources available to a given Internet user has become extremely large, and having an agent or set of agents to do the work of information retrieval for you is a very attractive idea. Agent systems currently operating, however, consist mostly of reasonably small systems suitable for a limited network area, but not for the entire number of information sources existing on the Internet.

Integration is a problem for information agent systems:  how do you start and keep an agent running for every database in the world, and perhaps a user-agent for every user, and then ensure that they can all communicate with one another?  Web search agents are encumbered by similar problems and by the Web itself.  Because Web tags provide little information about the content of Web documents[30], an autonomous agent must jump through several hoops in order to locate pages with desired information.

The use of information agents has become mainstream enough, however, that DARPA (the Defense Advanced Research Project Agency) has been working on industry standards for agent communication and knowledge transfer[32].  It has developed the Knowledge Query and Manipulation Language (KQML) for communication and a Knowledge Interchange Format (KIF) for knowledge transfer between agents.  This will be a large step toward the integration of separate agent systems, allowing them to communicate and relay information to each other.

The future of information agents in database information retrieval and in Web search is promising.  There is no getting around the fact that there are simply too many sources out there for a person sitting at his/her computer to sift through in search of specific information.  It is much easier to submit a query to an agent and let it find the information you need.  This saves both time and frustration in following links across the Internet, and it promises a bright future for intelligent search agents.

**ACKNOWLEDGMENTS**

We thank Andrew Williams for assistance in acquiring information concerning search agents on the World Wide Web.  This work is sponsored by NSF Awards IRI-9409263 and CDA-9401021.

## REFERENCES


1. D. Eichmann, “Agents,” talk given at the WWW Seminar, part of Uniforum ‘95, Dallas, TX, 1995.

2. Alta Vista Home Page, http://www.altavista.digital.com/, 1996.

3. O.A. McBryan, "GENVL and WWWW: Tools for taming the Web," *Proc. First International World Wide Web Conf.* Geneva, 1994.

4. C.L. Borgman, "Why are online catalogs hard to use?" *Journal of the ASIS*, Vol. 37, No. 6, 1986, pp. 387-400.

5. C.H. Fenichel, "Online searching: measures that discriminate among users with different types of experience," *Journal of the ASIS,* Vol. 32, No. 1, 1981, pp. 23-32.

6. "Intelligent Information Retrieval: An Introduction,"* Susan Gauch, *Journal of the American Society of Information Scientists, Vol.* 43, No. 2, March 1992, 175-182.

7. A.S. Pollitt, "CANSEARCH: An expert systems approach to document retrieval," *Information Processing and Management*, Vol. 23, No. 2, 1987, pp. 119-136.

8. P.J. Smith, S.J. Shute, D. Galdes, and M.H. Chignell, "Knowledge-based search tactics for an intelligent intermediary system," *ACM Transactions on Information Systems*, Vol. 7, No. 3, 1989, pp. 246-270.

9. S. Gauch and J.B. Smith, "An expert system for information retrieval," *Journal of the ASIS*, 44(3), 1993, pp. 124-136.

10. W.B. Croft and R.H. Thompson, "$I^3R$: A new approach to the design of document retrieval systems," *Journal of the ASIS,* Vol. 38, No. 6, 1987, pp. 389-404.

11. S.H. Myaeng and E.D. Liddy, "Information Retrieval with Semantic Representation of Texts," *Proc. Second Annual Symp. on Document Analysis and Information Retrieval*, 1993, pp. 201-215.

12. Gauch, S. & Wang, J. (1996) Automatic Word Similarity Detection for TREC 5 Query Expansion. In *5th Text Retrieval Conf. (TREC-5).* Gaithersburg, MD, (to appear).

13. Schütze, H. & Pedersen, J. (1994). A Cooccurrence-Based Thesaurus and two Applications to Information Retrieval. In *Intelligent Multimedia Information Retrieval Systems RIAO '94,* New York, NY, pp. 266-274.

14. Xu, J. & Croft, W.B. (1996). Query Expansion Using Local and Global Document Analysis, In *15th Ann. International ACM SIGIR Conf.*, Zurich, Switzerland, ACM Press, pp. 4-11.

15. *Foundations of Distributed Artificial Intelligence*, G. M. P. O'Hare and N. R. Jennings, eds., John Wiley & Sons, Inc., 1996.

16. M. Wooldridge and N. Jennings, “Intelligent Agents: Theory and Practice”, *Knowledge Engineering Review*, vol. 10, no. 2, June 1995.

17. E. Rich and K. Knight, *Artificial Intelligence*, D. Shapiro and J. Murphy, eds., 2nd Edition, McGraw-Hill 1991.

18. M. Genesereth and S. Ketchpel, “Software Agents”, *Communications of the ACM*, vol. 37, no. 7, July 1994, pp. 48-53.

19. *Blackboard Architectures and Applications*, V. Jagannathan, R. Dodhiawala, and L. Baum, eds., Academic Press, Boston, 1989.

20. A. Lingnau and O. Drobnik, “An Infrastructure for Mobile Agents: Requirements and Architecture”, *Proceedings of the 13th DIS Workshop*, Orlando, Florida, Sept. 1995.

21. C. Knoblock, Y. Arens, and C. N. Hsu, “Cooperating Agents for Information Retrieval”, *Proceedings of the Second International Conference on Cooperative Information Systems*, 1994.

22. M. P. Papazoglou, S. C. Laufmann, and T. K. Sellis, “An Organizational Framework for Cooperating Intelligent Information Systems”, *International Journal of Intelligent and Cooperative Information Systems*, vol. 1, no. 1, 1992, pp. 169-202.

23. T. Oates, M.V. Nagendra Prasad, and V. Lesser, “Cooperative Information Gathering: A Distributed Problem Solving Approach”, University of Massachusetts Technical Report CSTR-94-66.

24. K. Decker, V. Lesser, M.V. Nagendra Prasad, and T. Wagner, “MACRON: An Architecture for Multi-agent Cooperative Information Gathering”, *Proceedings of the CIKM Workshop on Intelligent Information Agents*, Baltimore, MD, 1995.

25. B. Sheth, “NEWT: A Learning Approach to Personalized Information Filtering”, MIT Media Lab, March 1996. Web site is http://agents.www.media.mit.edu/groups/agents/papers/newt-thesis/subsection2_5_2_1.html.

26. M. Huhns, U. Mukhopadhyay, L. Stephens, and R. Bonnell, “DAI for Document Retrieval: The MINDS Project, in *Distributed Artificial Intelligence*, edited by M. Huhns, Pitman Publishing, London, 1987, pp. 249-283.

27. E. Voorhees, “Agent Collaboration as a Research Discovery Technique”, *CIKM-94 Intelligent Information Agents Workshop*. InfoScout web site is http://www.info-scout.com/backup/info3c.html

28. D. Kuokka and L. Harada, “Supporting Information Retrieval via Matchmaking”, *Working Notes of the AAAI Spring Symposium Series*, pp. 111-11, Stanford University, March 27-29, 1995.

29. O. Etzioni and D. Weld, “Intelligent Agents on the Internet: Fact, Fiction, and Forecast”, *IEEE Expert*, vol. 10, no. 4, 1995, pp. 44-49. Softbot web site is http://www. cs.washington.edu/research/projects/softbots/www/softbots.html.

30. B. Doorenbos, O. Etzioni, and D. Weld, “A Scalable Comparison-Shopping Agent for the World-Wide Web”, University of Washington Technical Report TR96-01-03. Shopbot web site is http://www.cs.washington.edu/homes/bobd/shopbot-e.html

31. T. Oates, M. V. Nagendra Prasad, and V. Lesser, "Networked Information Retrieval as Distributed Problem Solving", *Proceedings of the CIKM-94 Workshop on Intelligent Agents*, 1994.

32. R. Patil, R. Fikes, P. Patel-Schneider, D. Mckay, T. Finin, T. Gruber, and R. Neches, “The DARPA Knowledge Sharing Effort: Progress Report”, *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, C. Rich, B. Nebel, and W. Swartout, ed., Morgan Kaufmann 1992.