

Active Information Gathering in InfoSleuthTM

M. Nodine, J. Fowler and B. Perry

MCC

3500 West Balcones Center Drive,

Austin, Texas 78759-6509 USA

email: {nodine, jfowler, bperry}@mcc.com

Abstract. InfoSleuth¹ is an agent-based system that can be configured to perform many different information management activities in a distributed environment. InfoSleuth agents provide a number of complex query services that require resolving ontology-based queries over dynamically changing, distributed, heterogeneous resources. These include distributed query processing, location-independent single-resource updates, event and information monitoring, statistical or inferential data analysis, and trend discovery in complex event streams. It has been used in numerous applications, including the Environmental Data Exchange Network and the Competitive Intelligence System.

1 Introduction

In the past 15-20 years, numerous products and prototypes have regularly appeared to provide uniform access to heterogeneous data sources. As a result, that access to heterogeneous sources is now taken as a “given” by customers. Current MCC studies indicate that, given the availability of products that achieve heterogeneous data access, new needs emerge for solutions to the following issues:

- Dealing with information at different levels of abstraction and in varying media forms.
- Fusing overlapping information from multiple sources into integrated wholes.
- Monitoring and reacting to changes, or patterns of changes, occurring across the networked information sources.
- Adapting to a changing environment with respect to data availability and domain coverage.

In other words, it is the active and integrated exploitation of information from these sources at appropriate levels of abstraction that is of real concern to applications of online information networks.

This paper provides an overview of the InfoSleuth project and its management of active information gathering and analysis tasks across heterogeneous

¹ The InfoSleuthTM Project (<http://www.mcc.com/projects/infosleuth>) is a R&D project at MCC that is supported by various industrial and government sponsors.

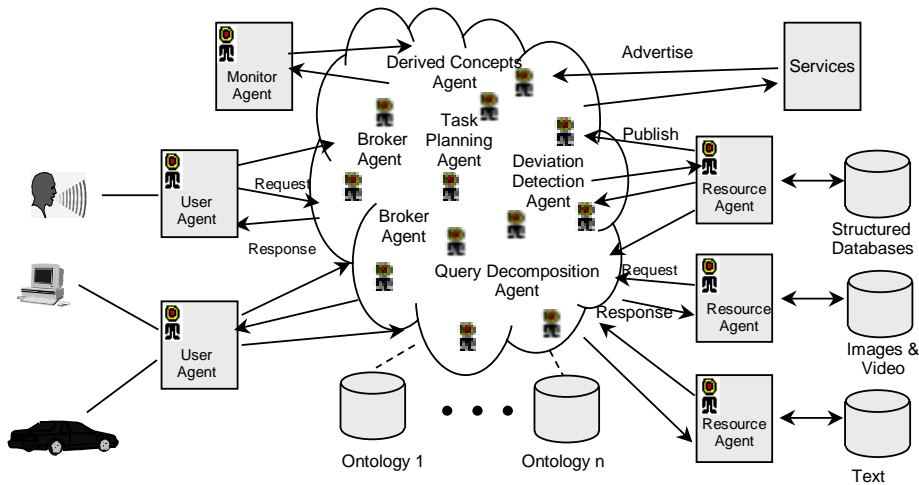


Fig. 1. Dynamic and Broker-based Agent Architecture.

information sources. It is an agent-based system that embodies a loosely coupled combination of technologies from information access, information integration, and information analysis disciplines. An agent system is a dynamic set of loosely interoperating (though cooperating), active processes distributed across a network or internet. Each agent in an information-gathering system is a specialist in a particular task in the information enterprise. Agents perform focused tasks for collecting data to create information at higher levels of abstraction or for collecting requests into generalizations of closely overlapping needs. Multiple agents interact and cooperate to solve complex information analysis tasks across multiple levels of abstraction. Typically an agent system uses some form of facilitation or market bidding to link up agents requesting services with agents providing services, and this capability is used to dynamically identify and compose goal-driven agent work groups.

We use the term “InfoSleuth” to refer both to an agent architecture for distributed information gathering and analysis and a deployed, advanced prototype implementation of that architecture. Figure 1 depicts the general model of the applications. An application domain is described by a set of *ontologies* that describe “domain objects, events and activities”. The agent system for the application consists of a network of agents capable of performing information routing, analysis, extraction, and integration. A network of external information sources contains data resources, at varying levels of abstraction, that provide partial evidence or facts for elements in the ontology. Resource agents wrap information sources, extraction their content, mapping it to one or more domain ontologies, and monitoring their information. Users interact with the system by engaging in a session of ontology-based requests with a user agent. Core agents are used “off-the-shelf”. The agent system dynamically identifies subsets of agents to interact and exchange information to best satisfy ontological re-

quests for higher-level (i.e., integrated and derived) information artifacts. This implementation is extended from that presented in [Bayardo et al., 1997], in the following ways: we currently use OKBC (not KIF) for communication about ontologies. We have a wider variety of information resources, including text, images, as well as object-oriented databases and file-system based data. New agents exist for multi-resource query decomposition and recomposition, concept derivation, value mapping, complex event monitoring, enforcement of business rules and deviation detection.

The remainder of this paper provides an overview of various InfoSleuth applications and the agents and agent interaction patterns that make these applications possible.

2 Example Application Areas

In this section, we present some of the application areas where InfoSleuth has been deployed, and discuss domain-specific features and types of requests that have been generated by users in these domains.

2.1 Environmental Remediation

The Environmental Data Exchange Network, or EDEN project, is a collaborative effort of several government agencies² to develop and demonstrate a means for sharing and using environmental data with other organizations [Pitts and Fowler, 1998; Fowler et al., 1999]. Currently, sharing environmental data is difficult because of the geographic distribution and heterogeneity of the information. The complexity of these systems often impedes access by secondary users, and their distribution frustrates attempts to draw data together to form a more comprehensive understanding of environmental conditions and actions.

The EDEN project is developing services that provide uniform access to specific geographically distributed environmental information databases through standard Web browsers. It focuses on understanding the metadata of environmental information and on sharing this information, but without incurring either the huge technical and political challenge of designing an integrated system among autonomous entities or the financial burden of maintaining redundant databases. EDEN uses InfoSleuth to retrieve information from distributed data sources, the current set of which is illustrated in Table 1. Among the resources employed is the Environmental Data Registry (EDR), an ISO/IEC 11179 metadata registry under development by the EPA for describing value domains, their various encodings, and translations between them. EDEN uses this resource for value mapping between some data elements found in the system. The Generalized Essential Multilingual Environmental Thesaurus (GEMET) will serve to standardize ontological terms as well as to provide potential translation capability among its several languages.

² Department of Defense (DOD), Department of Energy (DOE), Environmental Protection Agency (EPA). Funded through NIST contract 50SBNB5C9076.

CERCLIS	Oracle TM	EPA	Crystal City, VA	Superfund site profiles
Hazdat	Sybase TM	EPA/CDC	Atlanta, GA	Toxicology information
ITT	MS-Access TM	EPA	MCC, Austin, TX	Remediation technology
EDR	Oracle TM	EPA	MCC, Austin, TX	Environmental Data Registry
ERPIMS	Oracle TM	DOD	Brooks AFB, TX	Air Force site profiles
IRDMIS	Oracle TM	DOD	Aberdeen, MD	Army site profiles
ERIP	Oracle TM	DOE	Idaho Falls, ID	DOE site profiles
OREIS	Oracle TM	DOE	Oak Ridge, TN	DOE site profiles
Basel	MS-Access TM	EEA	MCC, Austin, TX	Basel Convention data

Table 1. EDEN Data Sources

The EDEN application is currently challenging InfoSleuth in various ways, including the following:

- Developers of the ontologies, or domain models, are challenged in their struggle to agree on a standard specification of the semantics of information for a diverse set of users drawn from multiple agencies and countries.
- Large amounts of real data must be integrated, forcing the agents to incorporate a variety of mapping facilities to map information onto ontological constructs.
- As different agencies represent their data differently, using different vocabularies specified in different lexicons, the system must incorporate value-mapping facilities to translate values from one lexicon to another. As lexicons are often large, these translation functions may not be easily embedded within the other agents.
- The target user community demands simple, yet expressive user interfaces. Query interfaces need to provide appropriate flexibility, while preventing the user from invoking queries that might consume large amounts of system resources to little benefit. Explanations of the derivations of results are required, especially in the face of seemingly incorrect behavior.

2.2 Competitive Intelligence and Technology Tracking

One of the services MCC provides is strategic technology monitoring and in-depth technical analysis of its member companies and their competitors, a practice known as *competitive intelligence (CI)*. The current “practice” is characterized by manual information gathering. It is estimated that CI professionals spend 80% of their time manually gathering data and only 20% of their time using tools to aid the gathering and analysis process. A particularly interesting and challenging application of InfoSleuth has been that of acquiring, integrating, and monitoring CI information from open sources. In general, CI is concerned with two basic activities:

1. Provide on-demand historical snapshots of competitor statistics across multiple behavioral indicators.

Source Category	Sample Sources
Macro-level statistics: sales, R&D expenditures, workforce profiles	SEC filings (www.sec.gov) Hoovers online (www.hoovers.com)
Current events	Press releases (www.prnewswire.com) CNN daily (www.cnn.com)
Growth technology	US, European, Japanese patent listings
Emerging technology	INSPEC publications database IEEE, ACM online bibliographies

Table 2. Classes of Competitive Intelligence Information Sources

2. Detect trends and shifts in trends for technology indicators for individual companies, for groups of related companies, or for general technology sectors.

These activities are performed with respect to four types of information products, as listed in Table 2. Note that, for the most part, these resources are either collections of semi-structured text documents, or structured records containing various free-text attributes. As a result, technologies for extracting semantic concepts from “textual structures” are of paramount importance in CI applications.

Given this brief review of CI needs and resources, we provide the following representative set of requests.

- Notify me when ‘electronic commerce’ technology exhibits a significant shift in ‘growth technology’ publications.
- Retrieve a comparative view of my competitors’ R&D spendings, workforce numbers, and patent awards.
- Retrieve the set of semiconductor patent codes Company X was associated with in 1994. Notify me when this set changes.
- Derive competitive associations between my competitors based on citation references in ‘emerging technology’ publications. Provide me with an updated derivation every 20 days.

These requests characterize the type of services CI professionals desire from information gathering and analysis tools. In all cases the requests require some level of extraction, integration, correlation, and monitoring of information from a network of open sources.

3 Agent Organization

Figure 2 shows the basic classification of the agent functionalities. The agents generally fall into three categories: user agents, resource agents, and core agents. **User agents** act on behalf of users to formulate their requests into a form

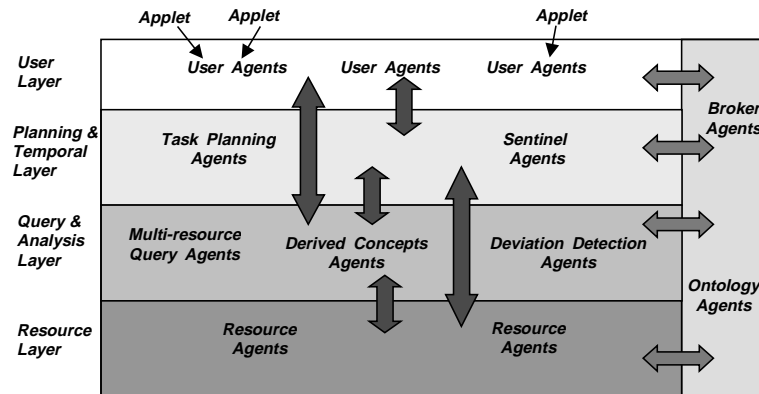


Fig. 2. Layers of Agents

understandable by the agents themselves, and transform results into a form accessible to the user.

Resource agents wrap and activate databases and other repositories of information. We consider any source of information a resource. Currently, we have implemented several different types of resource agents that can access different types of information. These include JDBC, text, flat files, and images. **JDBC resource agents** access relational databases via JDBC drivers. This enables us to integrate many different kinds of legacy relational databases. **Text resource agents** pull text documents indexed by one of several standard indexing tools, and apply focused syntactic and semantic language parsing to filter and extract information from them. **Flat file resource agents** access information stored in flat files that are parse-able using regular expression matching. Specialized **image resource agents** are tailored to specific sets of annotated images.

The core agents serve as the glue of the system, gathering information needed to process the users' requests, and synthesizing, filtering and abstracting that information into the level of abstraction that the user requires. We partition the core into service agents, query and analysis agents, and planning and temporal agents. Service agents, represented in the vertical box along the side of the figure, provide internal information to the operation of the agent system. Query and analysis agents, represented in the layer above the resource agents in the figure, fuse and/or analyze information from one or more resources into single (one-time) results. Planning and temporal agents, represented in the layer below the user agents in the figure, guide the request through some processing which may take place over a period of time, such as a long-term plan, a workflow, or the detection of complex events.

Service agents include broker agents, ontology agents, and monitor agents. **Broker agents** collectively maintain a knowledge base of the information the agents advertise about themselves. Brokers use this knowledge to match requested services with agents. Technically, the brokers collaborate to implement both syntactic and semantic matchmaking. When an agent comes on-line, it

advertises itself to the broker and thus makes itself available for use. When an agent goes off-line, the broker removes its agent from the knowledge base [Nodine et al., 1999]. **Ontology agents** collectively maintain a knowledge base of the different ontologies used for specifying requests, and return ontology information as requested. **Monitor agents** monitor the operation of the system.

Agents that do one-time query processing and/or data analysis include multi-resource query agents, deviation detection agents, and other data mining agents. **Multi-resource query agents** process complex queries that span multiple heterogeneous resources, specified in terms of some domain ontology. They may or may not allow the query to include logically-derived concepts as well as functions over slots in the ontology. **Deviation detection agents** monitor streams of data for instances that are beyond some threshold, where the threshold may be fixed or may be learned over time. Deviations themselves form an event stream for other agents to subscribe to.

Agents that do planning or processing over time include task planning and execution agents and sentinel agents. **Task planning and execution agents** plan how users' requests should be processed within the agent system, including how results should be cached. They may be specialized to particular domains, and support task plans tailored to their own domains. **Sentinel agents** monitor the information and event streams for complex events. A complex event is specified as a pattern of component events, which in turn may be other events such as changes in the information over time, triggers detected within individual resources, or deviations as detected by deviation detection agents.

4 Agent Functionality

4.1 Ontologies

Ontologies represent semantic concepts and terms familiar to the users in a particular domain. They represent knowledge about that domain, and are thus specified independently of the actual structure of the data and information in that domain. All agents use ontologies to communicate with one another; thus all agents are communicating at the same level of semantics as the user uses for his interactions.

We represent several types of information in an ontology:

- Entities, attributes and relationships.
- Class-subclass relationships.
- Lexicons or references to standard lexicons.
- Constraints on attribute types, cardinalities and values.
- Widely-useful derived concepts, computable from the underlying entities, attributes and relationships.
- Relevant statistical summaries.
- Composeable event types.

All mapping between ontologies and the underlying local terms and syntax of a data source is done by resource agents (described in Section 3). A resource agent encapsulates an information source, and presents that information to the agent-based system in terms of one or more ontologies. Thus, all communication with resource agents is done in terms of the ontology.

Ontologies are specified in OKBC [Chaudhri et al., 1998], which uses object-oriented description logic as an underlying data model. Ontologies are stored in an OKBC server and accessed via *ontology agents*. These agents provide ontology specifications to users for request formulation, to resource agents for mapping and to other agents that need to understand and process requests and information in the application domain.

4.2 Brokering

Brokering is the dynamic location and recommendation of active agents relevant to a specific task. When a requesting agent has a specific task that needs doing, and it must locate some agent that can do that task, then it asks the broker to recommend specific agents to which it can forward its request.

The “infosleuth ontology” is a special ontology used by agents to specify advertisements and queries. Concepts represented within the infosleuth ontology include:

Content - What ontology subset is accessible by the agent?

Services - What can the agent do for you?

Performance - How well can the agent do this at the moment?

Properties - Where is the agent? What protocols does it speak?

The brokering process follows an advertising/querying paradigm. Agents advertise the information they can provide and their capabilities to the broker in terms of constraints over the infosleuth ontology. Agents requesting services formulate queries for servicing agents to the broker in terms of constraints over the infosleuth ontology. The broker then uses constraint-based reasoning to find servicing agents whose advertisements match the constraints specified by the requesting agent. (We call this *semantic brokering*.) Finally, the broker returns a recommendation containing those servicing agents to the requesting agent.

For example, consider the query “List the names and products of technology companies with sites in Central Texas.” Consider that we have a set of resources that have advertised that they maintain company profiles, and another set of resources that have advertised that they have company site information, each advertising a select set of cities its information covers. The broker would filter through the resources with sites and identify those that, based on their advertised value constraints, could be in Central Texas. A second query to the broker would cover company profiles for technology companies, and specifically within the profile, any resources that have the company names and products parts of the profile. We use *constraint matching* to identify the most relevant agents that apply to a query expression.

4.3 Query Processing

Query processing involves receiving a query, discovering which resources have the information requested by the query, forwarding fragments of the query to those resources, and fusing the results into a single result that answers the query. The focus of the query processing task is the multi-resource query agent. Given a query over some domain ontology, the multi-resource query agent decomposes the query into:

1. A collection of queries $\{Q_i\}$ over individual classes in the ontologies, with the associated constraints over those individual entities, and
2. A global query Γ over the results of those queries (i.e., an integration plan), which generates the requested result. Note that the integration plan may involve joins or statistical functions.

Once it has decomposed the query, the query agent requests the broker to recommend resources that overlap queries in $\{Q_i\}$. A resource recommended by the broker is guaranteed to provide partial information to at least one query in $\{Q_i\}$ without violating the constraints in that query. The multiresource query agent then generates resource-specific queries based on the queries $\{Q_i\}$ to the individual resources, and collects the results in a temporary space to which the global query/integration plan Γ is applied. This interaction pattern is shown in Figure 3 and described in detail in [Perry et al., 1998].

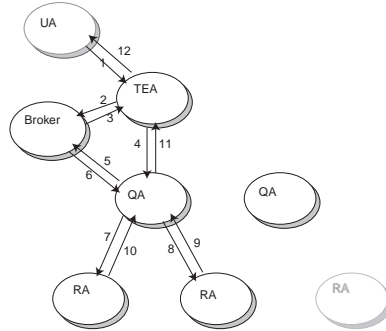


Fig. 3. Agent interaction pattern for query processing.

4.4 Requests over Time

Subscriptions and periodic requests monitor a set of information, specified as a query, over time. In a periodic request, the user specifies a query and an interval, the agents compute the answer to the query at the beginning of each interval, and forward each result to the user. This implements a classic pull interface. In a subscription/notification, the user specifies a query, and the agents immediately

generate a response to that query. When the answer to that query changes for some reason, the agents compute the specifics of the change, and forward the changes to the user. This implements a combination pull/push interface, where the user tailors the query to his needs (“pull”), but the changes are sent back to the user as needed only (“push”). The subscription interaction pattern results in subscription subqueries being propagated to a number of resource agents. Each of the resources, when discovering an event relevant to the subquery, will notify the task planning and execution agent, which then derives a new integrated snapshot of the overall query being monitored. This process is described in Figure 3.

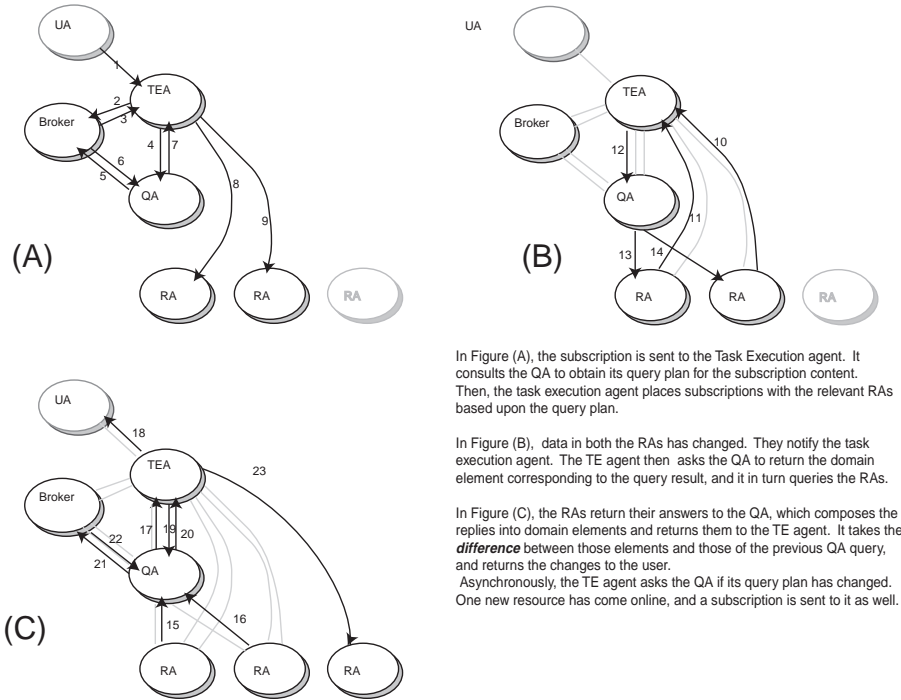


Fig. 4. Agent interaction pattern for subscriptions.

InfoSleuth also supports the subscription retrieval of information at higher levels of abstraction and aggregation than are necessarily represented in the underlying data. Information is aggregated and abstracted using a variety of data mining and complex event detection agents which work in a cascading manner to process and digest the information into a level of abstraction appropriate to the user. Currently, our derived concept, deviation detection and sentinel agents, as well as our planned association rule mining agent, collaborate to accomplish these aggregation and abstraction tasks. These processes are described in more detail in [Unruh et al., 1998].

Agent System	Ontologies	Semantic Brokering	Query Processing	Subscriptions	Event Detection
ARPI [Pastor et al., 1997]		✓	simple		
OOA [Martin et al., 1997]	✓	✓			
SIMS [Arens et al., 1996]	✓		complex		
Infomaster [Geddis et al., 1995]			complex		
RETSINA [Decker and Sycara, 1997]	✓	✓	simple		
InfoSleuth TM	✓	✓	complex	✓	✓

Table 3. Agent and agent-like information gathering systems, and their comparative functionality.

5 Related Work

There are a number of other agent systems targeted towards information gathering tasks, which share similarities with InfoSleuth in the functionality of the agents as well as the system organization. Most contain agents which “wrap” resources and combine information. None provide the same set of functionality with respect to ontologies, semantic brokering, heterogeneous and global query processing, aggregate subscriptions over multiple sources, and complex event detection. Table 3 provides a comparison of some of these systems.

6 Conclusion

In this paper, we have described InfoSleuth, an agent-based system for information gathering and analysis. The paper has emphasized InfoSleuth’s ability to extract and advertise information about semantic concepts; to integrate information from heterogeneous sources; and to provide long-running information-gathering and analysis tasks. Software agents are used to dynamically couple these capabilities together to support monitoring and analysis of ontological concepts which change over time, at multiple levels of abstraction. Emergent from these capabilities are a set of goal-driven agent interaction patterns.

InfoSleuth is a deployed, advanced prototype system performing information gathering and analysis over open information sources. The success of our approach has been to realize that *real information gathering applications require a goal-driven interaction between information access, information integration, and information analysis technologies*. Whereas each of these technologies has received much prior attention, a flexible integration of these disciplines has not yet occurred. This has resulted in relatively few successful deployments of information gathering technologies in open information networks. A major component of our ongoing work is the active deployment of the system to distributed

information gathering applications, such as EDEN and competitive intelligence. As these application deployments progress, we believe cooperating agent technologies will emerge as the proper technology to address these challenging, yet practical, application environments.

References

- Arens, Y., Knoblock, C.A., and Shen, W. (1996). Query reformulation for dynamic information integration. *Journal of Intelligent Information Systems*, 6(2):99–130.
- Bayardo, R. et al. (1997). InfoSleuth: Agent-based semantic integration of information in open and dynamic environments. In *Proc. ACM SIGMOD Int'l Conference on Management of Data*, pages 195–206. ACM Press.
- Chaudhri, V.K., Farquhar, A., Fikes, R., and Karp, P.D. (1998). Open knowledge base connectivity 2.0. Technical Report KSL-98-06, Stanford University.
- Decker, K. and Sycara, K.P. (1997). Intelligent adaptive information agents. *Journal of Intelligent Information Systems*, 9(3):239–260.
- Fowler, J., Nodine, M., Perry, B., and Bargmeyer, B. (1999). Agent-based semantic interoperability in InfoSleuth. *Sigmod Record*, 28.
- Geddis, D., Genessereth, M., Keller, A., and Singh, N. (1995). Infomaster: a virtual information system. In *Proc. ACM CIKM Intelligent Information Agents Workshop*.
- Martin, D. L., Oohama, H., Moran, D., and Cheyer, A. (1997). Information brokering in an agent architecture. In *Proc. Int'l Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*.
- Nodine, M., Bohrer, W., and Ngu, A. (1999). Semantic multibrokering over dynamic heterogeneous data sources in InfoSleuth. In *Proc. Int'l Conference on Data Engineering*.
- Pastor, J., Taylor, S., McKay, D., and McEntire, R. (1997). An architecture for intelligent resource agents. In *Proc. Cooperative Information Systems*.
- Perry, B., Taylor, M., and Unruh, A. (1998). Information aggregation and agent interaction patterns in InfoSleuth. Technical Report INSL-030-98, MCC.
- Pitts, G. and Fowler, J. (1998). Collaboration and knowledge sharing of environmental information: The EDEN project. In *Proc. IEEE Int'l Symposium on Electronics and the Environment*.
- Unruh, A., Martin, G., and Perry, B. (1998). Getting only what you want: Data mining and event detection using InfoSleuth agents. Technical Report INSL-113-98, MCC.