Use of InfoSleuth to Coordinate Information Acquisition, Tracking and Analysis in Complex Applications

Technical Report MCC-INSL-008-00

Larry M. Deschaine, PE
Science Applications International Corporation
Suite 200, 360 Bay Street
Augusta, GA USA
(706) 724.5589
Larry.M.Deschaine@alum.mit.edu
www.saic.com

Richard S. Brice, Ph. D.
Marian H. Nodine, Ph. D.
Microelectronics and Computer Technology Corporation
3500 West Balcones Center Dr.
Austin, TX

Abstract

Today's world is characterized by accessibility to a wide variety of information sources. This provides both the luxury of being able to know and use more information and the problem of accessing it in the manner required by our work and our computer applications. Many applications, including simulation-type applications, are no longer hampered by the availability of data, but rather are concerned with the accessibility of that data.

We see three separate needs that an information-oriented infrastructure can provide to such applications. These are:

1. Similar information may exist in many places, but in incompatible forms or formats. Applications need to view it as if it were coming from a single source.
2. Information processing must integrate both computer-based applications and real machines, such as sensors, giving a uniform methodology to deal with all kinds of information sources and processes.
3. Applications often must track the changing state of the information to develop up-to-date information feeds, summaries and analyses.

InfoSleuth employs intelligent agent technology to provide for integrated concept-based access to and awareness of unstructured, heterogeneous, and distributed information in a dynamically changing network of servers, data-collecting machines, and the World Wide Web. InfoSleuth agents can leverage existing legacy information resources, obviating the need for migrating information out of those sources and into new technologies. InfoSleuth agents collaborate to combine speedy information retrieval tasks with more cumbersome analysis and processing tasks. Furthermore, InfoSleuth is especially capable of monitoring changing or continuous data sources and converting event patterns to the appropriate level of abstraction for the users of the system.

From the user perspective, InfoSleuth uses concept-based access to information sources, analysis tools, and machines. Concept-based access is far more accurate and focused than that provided by currently available technologies. With InfoSleuth, users receive information at the level of abstraction and integration appropriate to their task.

We believe the InfoSleuth technology will help an application to go far beyond the reach of current information gathering and analysis technologies by facilitating the defining and building up of new applications from diverse, existing components.

InfoSleuth Functionality

InfoSleuth[1-22] is a very powerful agent based software application that performs information retrieval and fusion, event detection, data analysis, knowledge discovery and trend analysis using existing databases or the internet as data sources. Summaries of the core functionality, representing important advantages of this technology, include:

- Concept-based access paradigm: InfoSleuth uses concept based addressing, rather than syntactic features such as keywords.
- User perspective: The InfoSleuth user formulates queries using his or her own vocabulary. An "ontology", in InfoSleuth terminology, is the vocabulary appropriate to the user's domain of interest, rather than the (often cryptic) local names and structures that happen to appear in any given database or text resource. Thus different users may use entirely different vocabularies to reference identical information, which in turn may be maintained under some schematic vocabulary unfamiliar to both users.
- User Power: The InfoSleuth user has at his or her disposal the expressive power of the standard structured database language SQL, rather than the limitations of structured keyword search.

Supplementing this, the user may interact with the system through any of a variety of commercial or user supplied graphical user interfaces.

- Information retrieval and fusion: InfoSleuth agents access and fuse information from a wide variety of types of information sources, including external machines, databases, text and image repositories and the World Wide Web.
- Monitoring capabilities: InfoSleuth gives the user, on request, dynamic focused notification as the world of data changes. The user only need specify the style of information to be monitored. InfoSleuth transparently maps this to event monitoring on the appropriate resources.
- Distributed processing: InfoSleuth processes data where the data is. It enhances efficiency by distributing the processing of queries and data manipulations among multiple agents, each responsible for some subpart of the entire world of information.
- Collaborative Processing: InfoSleuth Agents cooperate with each other by pooling their resources to answer complex queries.
- Dynamic Architecture: InfoSleuth agents can come and go, i.e. be initiated, killed or moved, and InfoSleuth increases (or degrades) gracefully, using whatever services are available through the currently available set of agents.
- Scalability: InfoSleuth is extensible to a changing distributed world of information under a paradigm similar to that allowing growth of the Internet.

Technical Discussion and Design Method: Agents, Clients, & Tools

InfoSleuth is designed as an agent-based, object-oriented system. It was designed using certain hierarchical methodologies so that users can create and modify InfoSleuth agents easily using a standard set of interfaces and underlying components.

The InfoSleuth system consists of agents, clients and tools. Clients are user interfaces built using a common API. Agents are designed as instances of a set of Java classes called the generic agent shell. Agents communicate via conversations (which are specified using finite state automata) using a language called KQML (Knowledge Query and Manipulations Language) co-developed by academic and commercial participants interested in forging a *de facto* standard agent communications language. Tools such as the ontology creation and maintenance tools are built independently and with no overriding architectural hierarchy or relationship.

Clients allow users to update or query information via InfoSleuth and/or define event streams that are dynamically instantiated. Clients can subscribe to complex events extracted from the event stream by application of an event algebra. Changes to data anywhere in the information space can be thought of as an event. Thus, any client can ask to be notified when any datum changes values, or when some formally specified combination of data values changes. Other events might include a specific user logging on, or more than some number of users logging on simultaneously, a new data source joining the system (or an existing one leaving) etc. Users may create applications using Info/Sleuth's SleuthClient application programming Interface (API) or using the supplied TQML (Template Query Markup Language) interface.

Agents initiate, translate, decompose, receive, and synthesize queries and data. They are the "engine" of the system. All of the agents are designed as instances of a generic agent shell and thus all contain a common set of capabilities. This fixed portion of the agent architecture includes a set of finite state automata-based conversations that describe the kinds of conversations that are allowed. These conversations do not specify which agents participate in a conversation, but only which types of agents can participate. Thus any agent who knows how to produce some part of the answers to a request may join a conversation, or choose not to do so.

Fundamental to the operation of this dynamic architecture is a set of broker agents that advise agents on how to locate other agents that provide required functionality to complete a particular task or subtask. The broker keeps track of the different agents in the system, making it an authority on who is out there. Other agents access the broker as they need to; thus, the brokers effectively facilitate the construction of dynamic, short-lived cooperative communities of agents to perform specific tasks. They also can balance application loads across similar agents.

Particular classes of agents extend the capabilities inherited from the generic agent shell to perform specific tasks. The list of agent classes that have been implemented and evaluated under the InfoSleuth R&D project is quite large and only a few will be described here. For descriptions of other agents, and more detail, visit the InfoSleuth web pages that can be found by asking through the MCC web site located at http://www.mcc.com/projects.

- Portal Agents provide an interface tailored to the user's needs and maintain a persistent state for the user so that requests can continue to execute and

return results even when the user has logged off the system.

- Resource Agents advertise the contents of the information resource that they manage, and translate between the InfoSleuth lingua franca and the language spoken locally by the information resource technology. Resource agents may interface to data, image, or document bases, information-gathering machines such as sensors and specialized machines, and information-generating programs such as operating systems or web crawlers.
- Broker Agents collectively maintain an awareness of the dynamically changing set of available agents, including the content of the currently-available information space. Brokers offer advice to other agents that are attempting to discover or monitor information or events.
- Query Agents accept queries from users or other agents and decompose them into subqueries that each target an individual resource agent, and integrate the results as requested. This provides complex, distributed query processing to the users.
- Subscription Agents accept monitoring requests from users or other agents and maintain an active watch over the requested data or event specifications.
- Analysis Agents monitor streams of data performing different types of analysis as needed by the application. For instance, Deviation Detection Agents monitor streams of data for significant variations from the normal values for that data, or from variations from the current trends in that data. They use thresholds that may have been statically specified, or learned over time.
- Control Agents execute complex, multi-step tasks, where each task can be a query, a subscription, or the running of an analysis task such as deviation detection.

Implementation Methodology

Implementation of several large domain specific applications using the InfoSleuth technologies has provided proof of the concept that agent-based systems can significantly reduce the time and cost of developing and deploying systems that access large amounts of distributed, heterogeneous information.

The first step in creating any new application is agreeing on the "domain of discourse" of the users which allows an ontology to be specified and implemented. Implementation of the ontology requires no programming as such, but only the specification of a structure similar to an entity-relationship diagram needs to be produced. Tools exist for easily converting the specification to an implementation usable by InfoSleuth.

Once the ontology(s) have been created and added to the system, it is necessary to create a mapping and advertisement for each of the information resources that will participate in the application. For databases, the administrator of each information resource uses a set of point and click tools to map (some or all) of the information to the ontology, and to create an advertisement of the information that the information resource claims to be willing to export. For other types of information resources, which have non-standard or legacy interfaces, this mapping requires some direct implementation over a Resource Agent Shell. InfoSleuth provides such implementations for operating system commands and for web crawlers.

The final step is creating a specialized user interface that meets the needs of the user community. Creation of these interfaces can be done with the InfoSleuth tool suite with only minor programming required. Legacy GUIs can often be interfaced to the InfoSleuth system.

Case Examples

The above discussions provide for the basic understanding of the concepts used in the InfoSleuth technology. Recall that in the abstract we mentioned three general problems with applications that require access to information. In this section we will examine three applications and how they solve each of the three problems. We will show that we can deal with integrating similar information stored in multiple incompatible forms using our Environmental Data Exchange Network application. We will show an application that connects to legacy machines, databases, and large analysis programs with the genome sequencing and analysis application. Finally, we will show how we provide timely updates of different information analyses in our business intelligence application.

Accessing Diverse Information in EDEN

InfoSleuth is being used as a significant component of the Environmental Data Exchange Network (EDEN). EDEN is a collaborative effort of the EPA, the DOD, and the DOE, along with the European Environment Agency (EEA).

Over the past several decades, many US and European governmental agencies have collected vast stores of environmental data that is monitored or regulated by the various agencies and which is used by government and non-government scientists alike

to conduct research on remediation and control techniques. This information is stored in hundreds of disparate repositories that are hosted by numerous vendor information technologies. The schema for these information repositories have been separately designed and implemented and usually disagree on information representation and access in ways that make it impractical for agencies, or even different sites that are part of the same agency to share access to vital information.

InfoSleuth was applied in a manner that has developed a distributed agent architecture that addresses the need for semantic interoperability among information sources and analytical tools within diverse application domains. The current EDEN pilot demonstration enables integrated access via web browser to eight different environmental databases provided by offices of these agencies located in several states and Europe.

The EDEN Project is addressing the following common set of needs across the four funding agencies:

- The reduction of the reporting burden imposed by the agencies on each other.
- The improvement of mission performance through inter-organizational information sharing.
- The capability to share the best available and most current information, reducing duplication of effort at individual sites.
- The enabling of users to simultaneously access information from multiple sources.
- Coordination at the level of a common vocabulary, not the end use or structure of information resources.

At the application level, EDEN users access the InfoSleuth system using a GUI interface to the Portal Agent. Queries specified within this interface are forwarded to one of the query agents. The query agent analyzes which information it needs from which resources, and forwards queries for that information as appropriate. It then assembles the responses from the resources and processes them into the form requested in the query. This result it forwards back to the user via the Portal Agent.

For EDEN, InfoSleuth provides for semantic interchange among users by allowing an application developer to express the concepts and relationships of the application domain in high-level terms that are then translated into the low-level types of database schemas or semantic analyses of text and image resources. At the system level, InfoSleuth employs accepted standards where possible, to simplify data interchange and communication among processes. This type semantic integration of information resources reduces the time and expense necessary to share information across government agencies.

Integrating Data, Machines, and Analysis Tasks in Genome Sequencing and Analysis

Under a joint project with the US Department of Agriculture (USDA), MCC is developing an application that supports the laboratory protocols required to capture, sequence, analyze, and compare genetic material taken from livestock with genetic samples stored in other genetic databases that are publicly available. The initial sequencing of the genetic material is provided using specialized sequencing machines that provide imaged sequence files. These files then must be analyzed to identify the sequence of bases (e.g. GATTCG…) in each image and each of these sequences is compared with other accessible gene sequences.

This application includes a component not yet found in the EDEN application described above. To develop the USDA application, all three of the steps described above are necessary, and in addition a "workflow" or "planning" element is required to enable the Control Agents to perform their tasks.

The USDA Project is providing the following benefits, in addition to those mentioned above:

- The integration of data-generating machine components into the set of available resources.
- The replacement of the human component formerly used in the process of sequencing the task through a set of information gathering and analysis steps.
- The control of the execution of specific steps to balance out the load on the systems the steps must run on.
- The maintenance of continuous operation despite occasional agent failure.

At an application level, the plans which control the USDA application require one or more InfoSleuth Resource Agents to monitor the directories into which the sequence machines deposit their output. When an InfoSleuth Resource Agent notices that new imaged sequence files have been added to the database, it extracts this information and forwards it to an InfoSleuth Control Agent.

The Control Agent plan for processing the information includes several steps to be executed either sequentially, or in parallel. A collection of applications specific to genome analysis are marshaled in a pipeline fashion, with the output of one becoming the input for the next. The InfoSleuth

Control Agent walks the data through the pipeline until at the last stage, the analysis is complete.

At this point the plan causes the system to search publicly available genetic information sources in order to compare the results of the analytic pipeline with human and mice genetic information. When a match is found, relevant information about the human or mice genes is returned to the USDA lab and mapped to the livestock genetic material that initiated the analysis. In this way, livestock genetic researchers can automatically exploit the vast stores of knowledge that have been acquired for human and mice genes and apply that knowledge to their livestock research.

Timely Updating in Business Intelligence

A particularly interesting and challenging application of InfoSleuth has been that of acquiring, integrating, and monitoring technical competitive intelligence (CI) information from open sources. A primary activity in the CI domain is to correlate information from open sources, discover trends and associations across these sources, and detect significant shifts in trends over time.

The business intelligence application addresses the following needs, in addition to those mentioned above:

- The ability to classify arbitrary text documents according to concepts within the ontology, at a content (as opposed to keyword) level.
- The abstraction of information from multiple sources through time into summaries and trends.
- The analysis of information with respect to known trends to discover deviations and aberrant changes.
- The ability to note specific combinations of events that may be indicative of deeper changes within a company.
- The active monitoring of selected areas of the World Wide Web.

Several categories of agents in InfoSleuth support this application. At the base level, InfoSleuth's resource and multi-resource query agents bring information products into the system by performing extraction of semantic concepts from resources (i.e., data objects) and integration of semantically annotated data from related sources. For example, InfoSleuth includes Text Resource Agents, which pull information from semi-structured text files based on focused syntactic and semantic language parsing.

At a higher level, a variety of agents support different data analysis functions. Subscription Agents monitor specific sets of data for changes. Text Classification Agents analyze the content of free-text documents and categorize them by subject matter. Deviation Detection Agents analyze information trends and note when something occurs that is not expected based on the current trends. Specialized Control Agents called Sentinel Agents look for prespecified combinations of events and notify the user when they occur. General Control Agents control the timing and pipelining of various analyses.

As a result, within the business intelligence application, we can answer the following example questions (assuming that you are interested in ‘text database’ and ‘electronic commerce’ technology):

- Information gathering: Notify me of technology announcements about ‘text databases’ featuring one of a given set of companies.
- Deviation analysis: Notify me whenever articles featuring ‘text database’ technology are reported more actively than normal, for a given set of companies.
- Correlated deviations: If a company’s ‘text database’ articles are reported more actively than normal, and the same company’s ‘joint partnership’ articles are reported more actively than normal, then notify me if these happen in the same month.
- Filtered events, data fusion: If 'text database' technology articles are reported more actively than normal for a given set of companies, then notify me if any of the primary companies of the ‘active articles’ of Request 5 has a workforce smaller than 1000 employees.
- Event cluster: Notify me if ‘text database’ technology announcements are reported more actively than normal, where the companies referenced by the deviation events all share a common competitor.

Summary

In this paper, we have presented an approach to solving several needs generic to complex applications that gather and use information in various ways. These needs include:

1. Management and fusion of similar / related data from disparate, independent sources.
2. Merging of information retrieval tasks with heavyweight processes that use or analyze information, and synthesize new information that can in turn be used as input to other processes.
3. Timely notification of significant events and/or changes in specific information.

The intelligent agent technology InfoSleuth and representative deployments provide a proof-of

concept for a system architecture that meets these needs. We described the environmental data exchange application, which transforms a set of Internet-accessible data sources into a unified database for answering environmental queries. We described the genome sequencing application, which pipelines the processing of information through a series of steps, from acquisition of information from specialized machines, to storing that information, to analyzing it and finally to comparing the results with existing, public, and related information sources. Thirdly, we described the business intelligence application, which monitors real-time information feeds, analyzing the information as requested by business intelligence experts. We believe that these needs are applicable to a wide variety of other applications, such as simulation.

References:

MCC InfoSleuth technologies have been developed jointly with ten large commercial sponsors and several government agencies over the past five years. As part of this R&D process, several choices have been explored for knowledge representation, advertisement, brokering, representing services, communications languages, etc. This reference section contains references to a substantial collection of InfoSleuth publications that are available at:

http://www.mcc/com/projects/infosleuth/publications/index.html.

1. "Incompletely and Imprecisely Speaking: Using Dynamic Ontologies for Representing and Retrieving Information", Chung Hee Hwang. In Proceedings of the Sixth International Workshop on Knowledge Representation meets Databases, July, 1999 pp. 14-20. (Also MCC Technical Report INSL-043-99.)
2. "The Identification of Missing Information Resources by Using the Query Difference Operator" Michael Minock, Marek Rusinkiewicz and Brad Perry. In Proceedings of the International Conference on Cooperative Information Systems, 1999.
3. "Constructing Robust Conversation Policies in Dynamic Agent Communities", Marian Nodine and Amy Unruh. In Proceedings of the Workshop on Specifying and Implementing Conversation Policies, 1999. (Also MCC Technical Report INSL-020-99.)
4. "Agent-Based Semantic Interoperability in InfoSleuth", Jerry Fowler, Brad Perry, Marian Nodine, and Bruce Bargmeyer. SIGMOD Record 28(1), March 1999.
5. "An Agent Infrastructure for Knowledge Discovery and Event Detection", Gale Martin, Amy Unruh, and Susan D. Urban. MCC Technical Report INSL-003-99.
6. "Expressing Composite Events in InfoSleuth", Susan D. Urban, Amy Unruh, Gale Martin, and Marian Nodine. MCC Technical Report INSL-131-98.
7. "A Tractable Query Cache By Approximation (Revised)", Daniel Miranker, Malcolm Taylor, and Anand Padmanaban. MCC Technical Report INSL-127-98.
8. "Template Query Mark-up Language (TQML) 1.0 (Revised)" Michael Minock and Jerry Fowler. MCC Technical Report INSL-122-98.
9. "An Overview of Active Information Gathering in InfoSleuth", Marian Nodine, Jerry Fowler, and Brad Perry, in Proceedings of the International Symposium on Cooperative Database Systems for Advanced Applications}, 1999. (Also MCC Technical Report INSL-114-98.)
10. "Getting Only What You Want: Data Mining and Event Detection Using InfoSleuth Agents", Amy Unruh, Gale Martin, and Brad Perry. MCC Technical Report INSL-113-98.
11. "Information Aggregation and Agent Interaction Patterns in InfoSleuth", Brad Perry, Malcolm Taylor, and Amy Unruh. In Proceedings of the International Conference on Cooperative Information Systems, 1999.
12. "Semantic Brokering over Dynamic Heterogeneous Data Sources in InfoSleuth", Marian Nodine, William Bohrer, Anne Hee Hiong Ngu . In Proceedings of the International Conference on Data Engineering, 1999. (Also MCC Technical Report INSL-100-98.)
13. "Agent Communication Languages for Information-Centric Agent Communities", Marian Nodine and Damith Chandrasekara. In Proceedings of Hawaii International Conference on System Sciences, 1999.
14. "Experience with the InfoSleuth Agent Architecture", Marian Nodine, Brad Perry and Amy Unruh. In Proceedings of AAAI-98 Workshop on Software Tools for Developing Agents, 1998.
15. "Semantic Integration of Information in Open and Dynamic Environments", R. Bayardo, W. Bohrer, R. Brice, A. Cichocki, G. Fowler, A. Helal, V. Kashyap, T. Ksiezyk, G. Martin, M. Nodine, M. Rashid, M. Rusinkiewicz, R. Shea, C. Unnikrishnan, A. Unruh, D. Woelk, In Proceedings of the SIGMOD International Conference on Management of Data, 1997. (Also MCC Technical Report INSL-088-96, October, 1996.)
16. "Facilitating Open Communication in Agent Systems: the InfoSleuth Infrastructure", M. Nodine and A. Unruh. In Singh, Rao, Wooldridge (Eds.), Intelligent Agents IV--Agent Theories, Architectures, and Languages: Proceedings of the Fourth International Workshop, Providence, Rhode Island, July 24-26, 1997, pp. 281-295.
17. "Modeling and Querying Textual Data using E-R Models and SQL", Vipul Kashyap and Marek Rusinkiewicz. In Proceedings of the Workshop on Management of Semi-structured Data, in conjunction with SIGMOD '97. (Also MCC Technical Report MCC-INSL-037-97, March, 1997.)
18. "The Role of Java in InfoSleuth: Agent-based Exploitation of Heterogeneous Information Resources", N. Jacobs and R. Shea. In IntraNet96 Java Developers Conference. (Also MCC Technical Report MCC-INSL-018-96.)
19. "InfoSleuth Agents: The Next Generation of Active Objects", D. Woelk, M. Huhns and C. Tomlinson. In Object Magazine, July/August, 1995 (Another version is in MCC Technical Report INSL-054-95, June, 1995.)
20. "Carnot and InfoSleuth: Database Technology and the World Wide Web", D. Woelk and C. Tomlinson. In Proceedings of the SIGMOD International Conference on the Management of Data, May, 1995.
21. "InfoSleuth: Networked Exploitation of Information Using Semantic Agents", D. Woelk and C. Tomlinson. In Proceedings of the COMPCON Conference, March, 1995.
22. "Intelligent Search Management via Semantic Agents", D. Woelk and C. Tomlinson. In Proceedings of the Second International World Wide Web Conference, October, 1994.