Barton Stabler/SIS

Uncovering the next generation of active objects

Software agents are already being deployed and tested in real-world applications.

Darrell Woelk, Michael Huhns, and Christine Tomlinson

REQUIREMENTS FOR A NEW CLASS OF APPLICATIONS

A new class of applications is evolving thanks to ongoing advancements in computer hardware and software. These applications have the following attributes:

- They solve a specific business problem by providing a user with seamless interaction with remote information, application, and human resources.
- The identities of the resources to be used are mostly unknown at the time that the application is developed.
- The pattern of interaction (workflow) among the resources is a critical part of the application, but the pattern might be unknown at the time the application is developed and might vary over time.

The development of these new applications requires improved programming languages and improved system services. These should not be viewed as alternatives to such capabilities as OMG CORBA and OSF DCE, but rather as advanced features implemented at a higher level of abstraction and useful across multiple heterogeneous distributed computing environments.

Because each application executes as a set of geographically distributed parts, a distributed object architecture is required. It is likely that the objects taking part in the application were developed in various languages and execute on various hardware platforms. A simple, powerful paradigm is needed for communications among these heterogeneous objects. Due to the distributed nature of the application, an object may not always be available when it is needed. For example, an object executing on a PDA may be out of physical communications with the rest of the application.

Because the identities of the resources are not known when the application is developed, there must be an infrastructure to enable the discovery of pertinent objects. Once an object has been discovered, the infrastructure must facilitate the establishment of constructive communication between the new object and existing objects in the application.

Because the pattern of interaction among the objects is a critical part of the application and may vary over time, it is important that this pattern (workflow) be explicitly represented and available to both the application and the user. When an object has been discovered to be relevant to an application, the language for interaction and the pattern of interaction with the object must be determined. This interaction becomes part of the larger set of object interactions that make up the application. The objects collaborate with each other to carry out the application task.

AGENT-BASED SOFTWARE

A wide variety of software programs have been developed recently that are classified as software agents.[1,2] For our purposes, we will define two major classifications. The first type of agent-based software is focused on the interaction between a user and a computer. These agents monitor user interaction, initiate communication with the user, and may proactively initiate tasks on behalf of the user. The agent becomes more effective at assisting the user

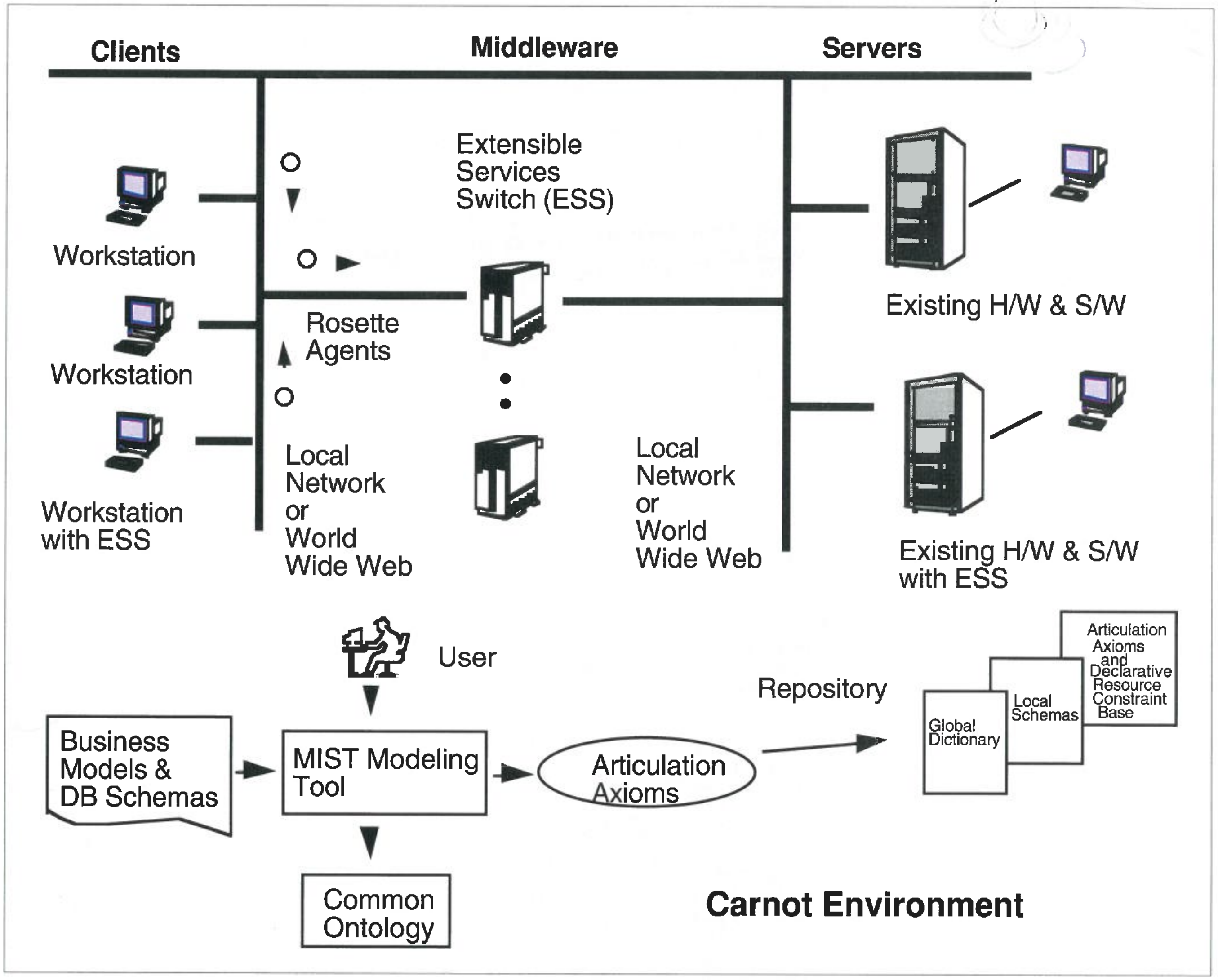


Figure 1. Carnot environment.

as it learns the user's habits and preferences. Examples of this research by MIT, AT&T Bell Labs, CMU, etc can be found in Riecken.[1]

The second type of agent-based software is focused more on the interaction among computing agents. The basic issues addressed are concerned with interoperability among geographically distributed agents executing on heterogeneous hardware platforms. There are two different approaches regarding a communication language for these agents. The procedural scripting approach causes execution of a remote task by sending a procedural script for interpreted execution at the remote site. Examples of this approach are General Magic Telescript and Tcl.[2] The declarative approach takes the view that only a declarative description of the task should be sent to the remote site. An example of this approach is ACL.[3]

The MCC InfoSleuth consortial project is conducting research on agent-based software and specifically is developing the second type of agents described above. Both procedural scripting agents and declarative agents are under development. In particular, an agent-based infrastructure is being developed to address the requirements for distributed applications described above.

The InfoSleuth research is building on the base technology developed in the recently completed MCC Carnot research project. Before describing the InfoSleuth project in more detail, we will first review the architecture and accomplishments of the Carnot project.

MCC CARNOT PROJECT

The Carnot project[4] was initiated in 1990 to address the problem of logically unifying physically distributed, enterprise-wide, heterogeneous information. A prototype has been implemented that provides services for enterprise modeling and model integration to create an enterprise-wide view, semantic expansion of queries on the view to queries on individual resources, and inter-resource consistency management. Carnot also includes technology for 3D visualization of large information spaces, knowledge discovery in databases, and software application design recovery. The prototype software has been used by the sponsors of the project to develop a number of applications, including workflow management, heterogeneous database access, knowledge discovery in large databases, and integrated access to both text and structured databases from a single initial query.

The implementation of the Carnot system has required unique advances in two technology areas. First, innovative techniques for

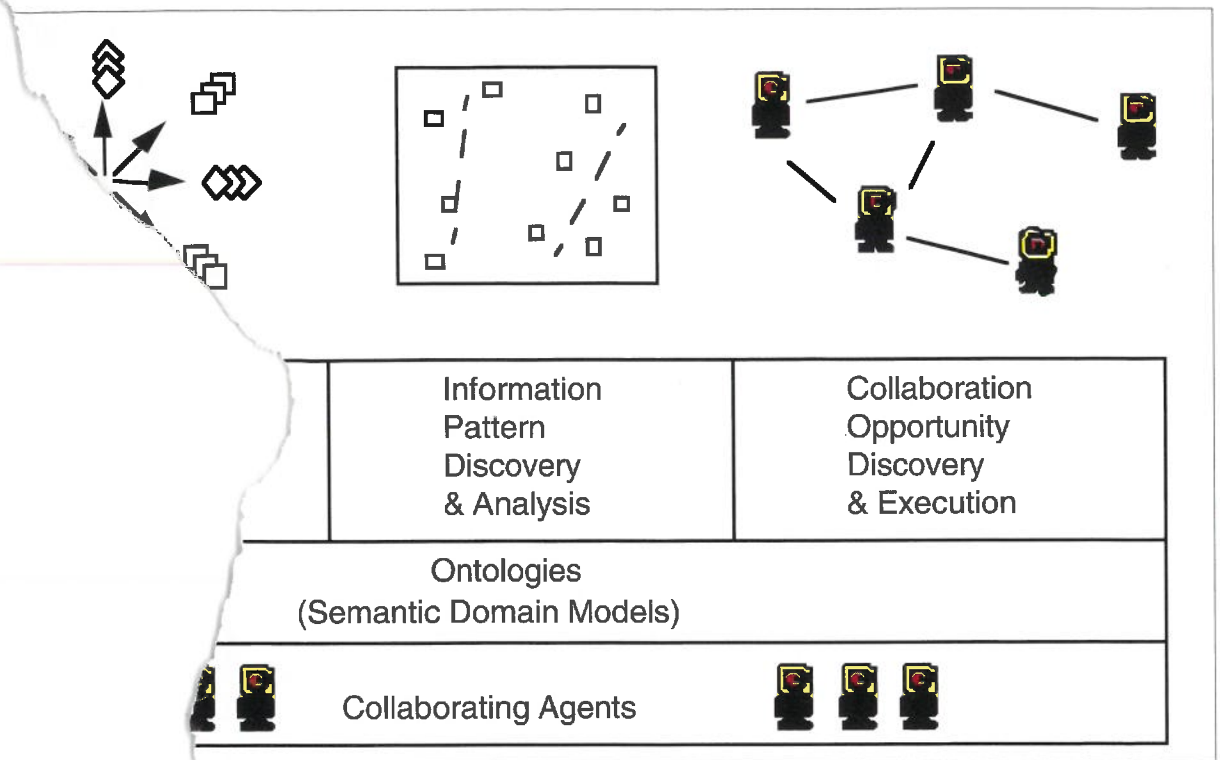


Figure 2. InfoSleuth architecture.

›r model of computation is asyn-
›ong concurrently executing en-

›ple and powerful model for syn-
›erference among concurrently

›l languages with some pro-
›rently concurrent language
›ss actions that are inher-
›currently. If a set of op-
›the Rosette language, it
›reter.
›for the dynamic defini-
›l then provides type-
›have been compiled
›nvironment, mak-
›Rosette executes as
›tor executing as an
process.
›ause the execution
of the distributed
›rough a commu-
› space provides a
›and uses an ad-
›m in the UNIX
›trieval of mes-
to retrieve a
›to the re-

questor blocking until a message is deposited by another Rosette agent.

RAD agents (declarative rule-based). The InfoSleuth project will also utilize rule-based agents that communicate through declarative messages. These agents are based on the RAD agents developed at MCC and enhanced in the Carnot project.[10]

RAD agents have a high-performance forward and backward reasoning engine that uses Warren Abstract Machine technology, a frame system integrated into a typed unification algorithm with multiple inheritance, a distributed truth-maintenance system, and a contradiction resolution mechanism. The actual communication among the RAD agents is enabled by the interaction of Rosette agents within the distributed Rosette tree space.

The communication among RAD agents uses a speech-act foundation based on a proprietary set of performatives. These are being replaced with a standard set of performatives based on KQML. The KQML performatives will be forwarded through the Rosette tree space. For example, a RAD agent's proprietary distributed truth-maintenance system already performs the default reasoning necessary to implement the untell, flush, and trash performatives.

The design of RAD agents is based on our observations of how a user interacts with a single, conventional knowledge base. A user can query and make assertions to the knowledge base, to which the system responds with either an answer or an acknowledgment, as appropriate. As a side-effect of forward or backward inferencing, the system can independently inform or make a request of a user, who must answer all requests. Interactions can thus be either user-initiated or system-initiated.

Similarly, RAD agents communicate at a fundamental level by exchanging messages, ie, queries or assertions. Agents can be passive or active, ie, they either respond to questions and commands knowledge representation have been developed to capture and maintain an enterprise model and map operations between an enterprise model and the physical databases. Figure 1 illustrates the Carnot environment. The MIST tool, shown in the lower right-hand corner, is a graphical-based interactive tool that assists the user in the integration of each database with the common ontology, which serves as the enterprise model. The integration step results in the generation of articulation axioms that specify mappings between databases and the common model. These articulation axioms are stored in the Carnot repository.

The second unique advance is a flexible, dynamic, distributed processing environment consisting of intelligent, autonomous computing agents that enable the retrieval of enterprise information and control enterprise processes. For example, a client application on the left side of Figure 1 may specify a query with reference to the common model. An intelligent Carnot agent will receive the query, consult the repository to discover which databases should be accessed, and create other agents to execute this access. Each of these agents also is given the mappings necessary to translate information from an individual database into the correct format.[5] Client applications access the Carnot environment through either a PC Windows ODBC interface or a Unix X/Open SQL CLI interface. Various local corporate networks are utilized to connect hardware systems.

The Carnot technology is currently being commercialized and products will be available in late 1995.

MCC InfoSleuth PROJECT

The MCC InfoSleuth project[6]
will extend the research agend
InfoSleuth project will devel
architecture that simplifies
tributed applications. The
shown in Figure 2.

Collaborating agents

The lowest layer consist
on behalf of a user. As
foSleuth agents. The
oped using the Rose
agents have been de
for Design) rule-bas
the InfoSleuth rese
determine the opti
should be deploye
mentations with o

Rosette agents
mance interprete
ented mechanism
at MCC since 19
model and the
simplify the
Actor mod

or initiate dialogs with another agent. A problem is solved by a RAD agent as follows: at any given time there are a number of computational agents, each having its own domain-specific knowledge, and a number of users, each assisted by an interface. A user or other agent gives a problem to one of these agents, which either:

- solves the problem itself using local knowledge and internal reasoning mechanisms, or
- decomposes the problem into subproblems, distributes each of these subproblems to an appropriate agent, and then integrates the solutions returned by these agents. This cooperation enables the solution of the problem.

Ontologies (semantic domain models)

The ontologies layer of the InfoSleuth architecture in Figure 2 provides collaborating agents with a common vocabulary and a common semantic model for interaction in some application domain. For example, a set of collaborating InfoSleuth agents in a medical application may have access to a healthcare ontology. To revisit the example used earlier, there might be an agent A provided with rules that represent mappings between the healthcare ontology and two different hospital legacy databases. Agent B can request information from agent A in terms of the healthcare ontology. Agent A maps the query into the proper subqueries to access the two legacy databases, create agents to execute the access, and maps the resulting data into the proper format to be returned to agent B.

There are two important points to be made with respect to the two lower layers of the InfoSleuth architecture. First, the ontologies layer itself will be implemented as a set of collaborating agents, so that it is scalable and extensible. Second, as in the Carnot architecture, some applications may not require the use of an ontology. In these applications, either the individual agents share a common internal structure for information or they are programmed to translate from their internal information structure to the internal information structure of another specific agent.

The InfoSleuth project will investigate three categories of applications, as shown in the top layer of Figure 2: *information resource discovery*, *information pattern discovery & analysis*, and *collaboration opportunity discovery & execution*. The common theme of these application categories is that there is a discovery process that requires (or allows) interaction among human users and computing agents. Each discovery process must be explicitly represented so it can be documented, modified, reasoned about, and possibly aggregated with other discovery processes.

Information resource discovery

Information resource discovery applications have increased in visibility and importance with the explosive growth of the Internet and the World Wide Web. A distinguishing characteristic of these applications is that new information sources are constantly being added, and there is local autonomy, ie, no network-wide control of the registration of new information sources and their content.

In this type of environment, traditional techniques for expressing and optimizing database queries are inadequate because of the rapidly changing schema information and the fuzzy nature of the

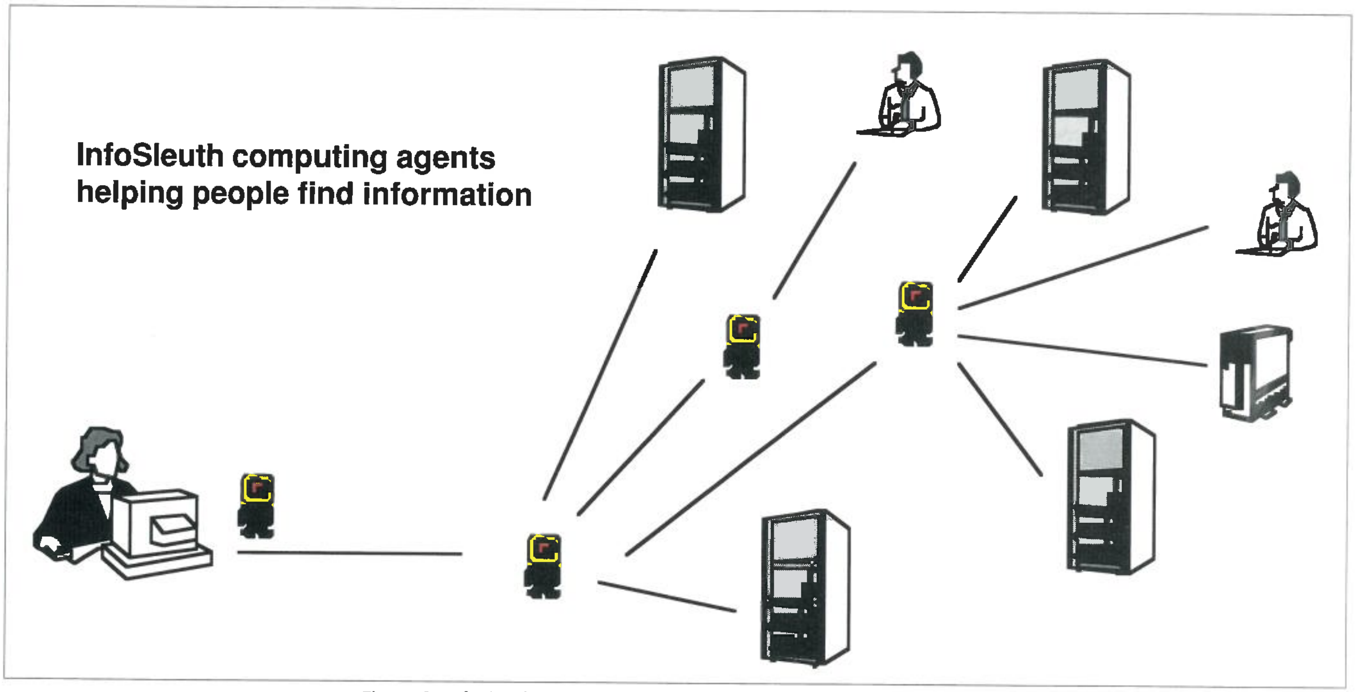


Figure 3. InfoSleuth computing agents helping people find information.

queries. Text search techniques and interactive navigation techniques are also inadequate because of the immense size and (potentially) remote distribution of the available information.

Research projects and commercial products related to the World Wide Web have focused on the development of spiders that traverse the URL references in World Wide Web pages and build a keyword index that can be used to find pages of interest. Examples of spiders are Lycos[11] and Webcrawler.[12] The Harvest system at the University of Colorado[13] and the WebAnts project at Carnegie Mellon University[14] are developing systems that utilize multiple coordinated spiders to traverse parts of the Web and merge the results into single or multiple indexes. The InfoSleuth project is investigating the use of InfoSleuth collaborating agents to both coordinate the traversal of the Web to build the index and coordinate the use of multiple indexes for responding to user queries.

The InfoSleuth project is also investigating techniques[15] to improve the indexing of information resources in the World Wide Web by using the ontologies layer of the InfoSleuth architecture:

1. Providers of information will advertise its availability by relating the information to the ontology.
2. Clients that are searching for information will discover the availability of potentially useful advertised information. Autonomous InfoSleuth agents will be deployed to search for information, remaining active to monitor for the addition of pertinent new information.
3. InfoSleuth agents will cooperate to integrate the information from the various resources.

The use of a common ontology by both providers of information and clients searching for information will enable an InfoSleuth agent for an information provider to search for clients that might be interested in the information.

Information pattern discovery & analysis

Once a pertinent information resource has been discovered, there is typically a phase in which its contents are analyzed in more depth. In the case of World Wide Web pages, this phase consists mostly of browsing pages and possibly creating a new hypermedia document with references to existing Web pages.

Agents may also assist in the discovery of patterns in the relationships among elements of information in an information resource. This is particularly true when the information resource is a structured database (a relational or object-oriented database, etc), but analysis of text, audio, image, and video databases is also possible.

This pattern discovery and analysis phase can also operate across multiple discovered information resources. It should be noted that using Carnot for enterprise information integration is a special case of information resource discovery followed immediately by information pattern discovery and analysis.

The InfoSleuth research is using the Logic Data Language (LDL++),[16] developed at MCC and enhanced in the MCC Carnot project, as a language for pattern discovery and analysis. LDL++ is a deductive database system that integrates concepts of logic programming, relational database technology, and object-oriented programming. LDL++ provides rule-based interaction with and integration of existing databases.

Collaboration opportunity discovery & execution

Agents can collaborate on tasks other than discovery and analysis of information. The InfoSleuth project will also investigate a broad range of other applications and develop prototypes for a few applications. These other applications can be characterized by the requirement to dynamically discover the requirement for collaboration and then dynamically discover the pattern of collaboration that should be followed.

EXAMPLE APPLICATION USING INFOSLEUTH AGENTS

An example application using InfoSleuth agents is illustrated in Figure 3. A marketing manager needs to determine the optimal geographical location for test marketing a new product. To make this decision, he/she will need the following types of information:

- Have other marketing managers in the company test-marketed similar products?
- Have competitors test marketed similar products?
- Have similar products sold well?
- What would be the best geographical location for a test market?

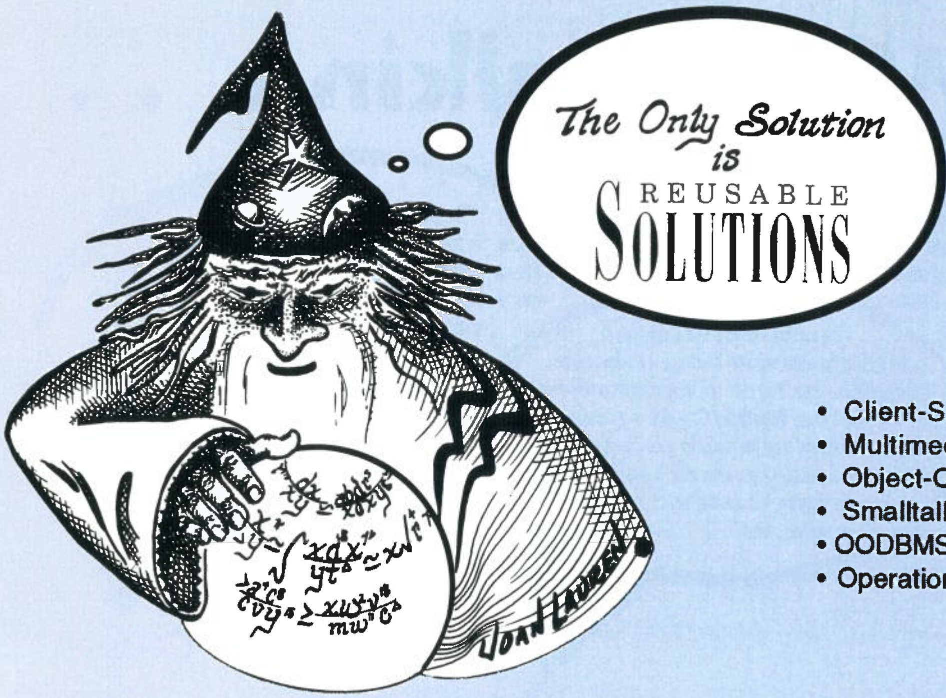
In the past, the marketing manager might have conferred with a marketing analyst, who would access information in a company database. But the analyst would also depend heavily on interaction with others to find needed market information. It was not necessary for the analyst to know of the existence of the databases containing the information and how to physically access the databases. The analyst only had to know someone who knew how to find it.

Each of the other people helping in this search for information had some specialized knowledge of an area, including knowledge of the specialities of other people. Two people could adapt their interactions to handle slight variations in the way questions were asked or confusion about the meaning of information that was found.

Today, as computers and communication among them have become more ubiquitous, more of the information the marketing analyst is seeking is likely to be accessible online. Finding it remains difficult, however, as the typical search techniques available today on the Internet and World Wide Web are static links and syntactic keyword indexes. People are still a part of this process, but they can be augmented and represented by InfoSleuth agents as shown in Figure 3.

InfoSleuth will provide software tools that assist people in expressing their knowledge about information sources in a clear and concise manner. An InfoSleuth agent can represent the marketing analyst assisting the marketing manager. The marketing manager requests information from the agent through a natural interaction. The agent then independently sets out to find the information, possibly returning to the manager for clarification or to report status. The agent also interacts with other InfoSleuth agents to find the proper sources of information, retrieve it, and then present it to the manager in an understandable manner. The marketing manager can request that the InfoSleuth marketing analyst agent continue to monitor information sources to identify changes in the requested information that might represent trends in the market.

The InfoSleuth marketing analyst agent might contact an InfoSleuth real estate investment agent to better understand changes in the real estate market in various geographical locations. The real estate investment agent may access information in a number of different online databases. Just as with a human agent, the Infosleuth agent will receive compensation for its assistance, as does the owner of the online database. A good real estate investment agent will have many customers and owners of databases will seek to have their databases accessed by that agent.



SUMMARY

The emergence of a ubiquitous worldwide communication infrastructure is enabling a new class of applications characterized by their requirements for intelligent collaboration among independently developed applications executing in a dynamically expanding, geographically distributed environment. The MCC InfoSleuth research project is developing agent-based infrastructure software that will meet these requirements. The software agents are being deployed and tested in real-world applications by MCC and the sponsors of the InfoSleuth project. ■

References

1. D Riecken. Introduction, COMMUNICATIONS OF THE ACM (Special Issue on Intelligent Agents) 37(7):18–21, 1994.
2. K Indermaur. Baby steps, BYTE, March, 97–104, 1995.
3. M Genesereth and S Ketchpel, Software agents, COMMUNICATIONS OF THE ACM 37(7):48–53, 1994.
4. http://www.mcc.com/projects/carnot
5. D Woelk et al. Using Carnot for enterprise information integration, PROCEEDINGS OF THE 2ND INTERNATIONAL CONFERENCE ON PARALLEL AND DISTRIBUTED INFORMATION SYSTEMS, January 1993, pp. 133–136.
6. http://www.mcc.com/projects/infosleuth
7. C Tomlinson et al. Workflow support in Carnot, DATA ENGINEERING BULLETIN 16(2):33–36, 1993.
8. G Agha. Concurrent object-oriented programming, COMMUNICATIONS OF THE ACM, September, 125–141, 1990.
9. N Carriero and D. Gelernter. Linda in context, COMMUNICATIONS OF THE ACM 32(4), 1989.
10. M Huhns and D Bridgeland. Multiagent truth maintenance, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS 21(6):1437–1445, 1991.
11. Lycos: http://lycos.cs.cmu.edu
12. WebCrawler: http://webcrawler.cs.washington.edu/WebCrawler/WebQuery.html
13. Harvest: http://harvest.cs.colorado.edu
14. WebAnts: http://thule.mt.cs.cmu.edu:8001/webants
15. D Woelk and C Tomlinson. InfoSleuth: Networked exploitation of information using semantic agents, COMPCON, March 1995.
16. K Ong et al. A deductive database solution to intelligent information retrieval from legacy databases, PROCEEDINGS OF THE 4TH INTERNATIONAL CONFERENCE ON DATABASE SYSTEMS FOR ADVANCED APPLICATIONS, April 1995.

Darrell Woelk is Project Manager of the Infosleuth Project, and Michael Huhns is a Senior Member of Technical Staff at Microelectronics and Computer Technology Corporation (MCC), Austin, TX. Christine Tomlinson is a Senior Scientist at Express Star Systems Inc, Austin, TX.

Pullout
The Summer 1995 Object Buyer's Guide
...Your source for products and services