Using Intelligent Agents to Combine Heterogeneous Distributed Data

Marie M. G. DeRico
dm9202@exmail.usma.edu

Ronald B. Byrnes, Jr.
d9148@exmail.usma.edu

Joseph H. Schafer
dj4149@exmail.usma.edu

John A. Marin
dj7900@exmail.usma.edu

Office of Artificial Intelligence, Analysis, and Evaluation
United States Military Academy
West Point, NY 10996

Michael D. McNett
dm8035@exmail.usma.edu

Department of Electrical Engineering and Computer Science
United States Military Academy
West Point, NY 10996

George F. Stone III
George_Stone@stricom.army.mil

US Army Simulation, Training and Instrumentation Command
Orlando, FL 32826

ABSTRACT

Obtaining data from the World Wide Web has become second nature to most users. In many cases, the sources of this data are distributed databases. Effective interaction with distributed databases means providing the user with location transparency. That is, it should appear to the user as if the data is located at a single site. One major challenge in obtaining location transparency is the integration and presentation of heterogeneous, distributed data in near-real time with a user-friendly, web-based interface. One promising approach to achieve this goal is through the use of intelligent agents.

This paper will describe two ongoing projects from the Office of Artificial Intelligence, Analysis, and Evaluation at the U.S. Military Academy, West Point, New York, that apply intelligent agents to the problem of integrating and presenting heterogeneous, distributed data. The first project is a web-based, simulation scenario-generating application that integrates data from distributed sources to build a simulation scenario for an interactive wargaming simulation called Janus. This scenario-generating application is called Dynamic Scenario Builder (DSB) and will ultimately integrate data from the Army’s Functional Description of the Battlespace (FDB) with Janus or other simulation databases to generate simulation scenarios. The second project involves the development of a system to enable logistics planners to access distributed databases from remote locations and autonomously update data sources such that modifications are dynamically mirrored and displayed to all users of the distributed system. A key feature of this research is the visual presentation of the summarized information in a meaningful manner such that “drilling down” to more detailed information is evident. This second project applies a platform for mediators called HERMES (Heterogeneous Reasoning and Mediator System) as a means of data integration from several heterogeneous distributed databases.

Key Words: intelligent agents, software agents, database, heterogeneous

1. INTRODUCTION

Management of heterogeneous distributed data in organizations has long been a challenge for systems developers. A group of prominent database researchers met, discussed, and published their collective opinion on future prospects concerning database research in February 1990 and again in May 1995 [14]. These researchers stated that significant database research problems included support for multimedia objects, distribution of information, and ease of database management and use [14].

One way to solve these perplexing problems is through the use of intelligent agents. The authors will define a set of guidelines that can be used to implement this type of solution on a broad scale by distinguishing between intelligent software agents and software agents that are used for data searches.

The nature of data fragmentation and a summary of the major issues involved are briefly presented. The concept of the intelligent agent is introduced and examined.
cases are described and analyzed where intelligent agents were used to implement partial or total solutions to fragmented data integration in real-time. Finally, conclusions and future research potential are discussed.

What is Distributed Data?

In the reference book, Database Systems: Principles, Design & Implementation by Catherine Ricardo [13], a distributed data system is described as “one in which multiple database sites are linked by a communications system in such a way that the data at any site is available to users at other sites”. A system such as this has several characteristics: (1) provides the user an interface that is transparent to where the data actually resides; (2) ability to locate the data; (3) a DBMS to process queries; (4) network-wide concurrency control and recovery procedures; and, (5) mediators to provide translation of queries and data between heterogeneous systems.

Communications in a distributed system is a complicated and rapidly changing field [13]. There are three basic components in any data communications system: the source, the medium and the sink. The message originates at the source; the path that the message travels is the medium; and the mechanism that presents the data to the user is the sink. There are many different links, lines, channels, or circuits over which the data can travel resulting in a complex communication medium. In addition, there are many characteristics that must be considered when transferring the data: path establishment time, network delay, transfer rate, and reliability [13]. This brings us to the primary problem addressed in this paper—obtaining distributed data over the Internet and presenting this data in hypertext markup language (HTML) format in real-time [14].

2. DISCUSSION OF THE PROBLEM

The Web is an unstructured mass of information buried in documents designed for human consumption [2]. The unstructured nature of information is what makes retrieval and presentation a tough problem. There are differences in the relational schema (the ability to match data components from one table to another through a unique key that exists between two tables); the vocabulary (the use of different attribute names by different databases referring to the same concept); and semantics (the meaning of a query to the database). For example, a company needs to combine information from two different personnel databases, each with its own employee table. This type of problem could occur when companies merge and have a need to incorporate their data. One database uses LNAME in its structure as an attribute for the employee’s last name, and the other database uses LASTNAME in its structure as an attribute for the employee’s last name. Each database is relational and specific information needs to be obtained from each. The query must pull the data out of the relational databases using a unique key to associate the data. Potentially, each database has a different method of implementing the unique key to relate the data. The problem then becomes obtaining all the data needed to satisfy the user’s requirements.

The re-assembly of distributed fragments as mentioned above is challenging due to the heterogeneity of the data [6]. Another difficulty with heterogeneity is that programs are written by different people at different times. The developers use different languages and often provide different interfaces for users to manipulate. These programs are continually modified or replaced making retrieval and integration of data harder [7]. Also, the problem broadens when we desire to extract data from sources other than databases, such as text documents, web pages, and electronic libraries. The data from these sources can take the form of graphics, sound, animation, etc.

Data manipulation and placement of that data on the web is a fairly simple process. Implementing interactive Web-to-database applications is more complex [12]. The reason for this is the data is typically in different formats and maintained by a variety of sources. Some of these sources could be legacy systems, relational databases, object databases, flat files, text files, word-processing pages, and web pages. The problem increases when these sources have to be queried or updated.

Publishing data on the web can be done in three different ways. One of the simplest is the static web page—one that contains only text and graphics. This process consists of a web browser requesting an HTML page from the Internet. The Internet then returns the requested page to the desktop and displays the results [12].

The second method of data publishing is through the use of form pages. A form page is developed using basic HTML code and Common Gateway Interface (CGI) scripts. These CGI scripts respond to requests from a web client by executing a file that returns HTML as output. The browser requests a form page and presents it to the user. The user fills in the form and submits it through the browser to the Internet. A CGI script takes the information filled in on the form, processes it, then returns the results to the desktop [12].

The third method of data publishing is through a database-driven web page. This method is a simple query page where the user fills in some parameters on a form and submits it to the Internet. The query is run and the requested data is returned to the desktop [12].

These processes are simplistic in their function. Of more interest is the situation where one requires selected data from heterogeneous data sources. Suppose the developer wants to integrate data from multiple data sources and place them in a web application in real-time? According to Rennhackkamp in his article “Desegregating DBMSs”, to integrate “data from various sources on dynamic, interactive, database-driven Web pages, one must implement a three-tier, four-tier or n-tier applications architecture” [12]. In a three-tier architecture the Web server makes a request to obtain data from a database server. This request is handled through a standard client/server connection. The database server then retrieves the data from a remote database through a database gateway and sends the data back. In a four- or n-tier architecture the application server requests the required data from various database servers and the database servers
retrieve the data as was described for the three-tier architecture above [12]. A developer can use a tool (such as an intelligent agent) that could perform the process of the application server to work with web-development tools.

3. INTELLIGENT AGENTS

Agents have been given many definitions by the researchers who work in the field. The definitions vary but the researchers generally agree that autonomy and ability to act on behalf of a user for a specific purpose is an indicator of agency [11]. The user can be a human, a program, or another agent.

An intelligent agent (IA) according to Pattie Maes, a noted researcher in the field of intelligent agents, is "a computer program to provide assistance to a user dealing with a particular computer application" [8]. The IA must be able to react and pro-act according to the user's need to perform a task, in many respects similar to a personal assistant. Some IAs may have the ability to "look over the user's shoulder" and detect patterns and regularities in the user's behavior [8].

Common Traits of Agents

Agents can be as simple as subroutines and as complex as an interactive personal assistant. According to Franklin and Graesser, an agent is "...a software-based computer system that enjoys the following properties" [5]:

- Autonomy: agents operating without the direct intervention of humans or other agents, while exhibiting control over their actions and internal state
- Sociability: agents collaboratively interacting with other agents (and possibly humans) via an agent-communication language
- Reactivity: agents perceiving their environment (which may be the physical world, a user via a graphical user interface, a collection of other agents, the INTERNET, or all of these combined) and responding in a timely manner
- Flexibility: agent actions do not have to be scripted; rather, agents can be easily modified to perform other tasks

Intelligent agents exhibit the following additional properties:

- Pro-activity: instead of simply reacting in response to their environment, intelligent agents are able to exhibit goal-directed behavior by taking the initiative.
- Adaptivity: intelligent agents have the ability to learn through inference and experience
- Mobility: in certain circumstances, intelligent agents may be capable of transporting themselves from one machine to another in order to perform a task

These attributes help distinguish between a standard software program and agents. "All software agents are programs, but not all programs are agents" [5].

How are intelligent agents developed?

Agent-based software engineering is an approach to software development whereby software applications are written specifically to function as software agents. This application communicates with its peers by exchanging messages in a communication language understood by each. This language is referred to as Agent Communication Language (ACL). Agent Communication Language consists of three parts: its vocabulary, an inner language called Knowledge Interchange Format (KIF), and an outer language called Knowledge Query and Manipulation Language (KQML). An ACL message is a KQML expression in which the arguments are terms or sentences in KIF formed from words in the ACL vocabulary. The vocabulary of ACL is listed in a dictionary facilitates human understanding of each word. The words are written in KIF so that agents can understand their meaning [7].

Real-Time Processing

The agent description given by Wooldridge and Jennings in their writings infers operation in a real-time environment [19]. A real-time system requires a response time that must be fast enough to control a process or query when needed. According to Jose C. Brustoloni, "...agents must be able to perform in the real world. Agents must be reactive— that is, be able to respond to external, asynchronous stimuli in a timely fashion" [1].

For example, a human resources (HR) manager may need information about an employee who applies for a specific job. This job would result in a promotion for the employee. The HR manager may have access to an intranet that connects several databases containing information about the position and the individual. A real-time process would use resources or an intelligent agent to pull current data from the distributed databases allowing the HR manager to quickly evaluate the candidate. Given this information, the HR manager can make an informed comparison to determine the best fit. This type of application may require some intelligence on the part of the agent or obtain access to an intelligent knowledge base that would make the match. The information would be current allowing the HR manager to make a better decision.

Some agents in the commercial community may operate in real-time, but these are typically in the form of commercial programs whose focus is the Web. These web-based agents are considered to be search utilities rather than intelligent agents [11]. The primary purpose of these web-based agents is to aid the user in searching for specific information. Because they follow a set of user specified rules to perform a certain function, there is no real intelligence involved [10].

4. APPLICATIONS OF INTELLIGENT AGENTS

Intelligent agents can be used to manage information overload. They have the ability to sort and filter enormous quantities of incoming data required to satisfy a user's goal. This ability will enable the user to obtain information in a timely and efficient manner. An IA can also enhance the user’s capability to make complex decisions based on the agent’s ability to obtain detailed information necessary to make an effective decision. The IA has the ability to do tedious, time-consuming, repetitive searches and retrievals through distributed databases independent of user
intervention [9]. In addition, IAs work quickly without complaint.

An added value of an IA is its ability to choose the facilities to be used to obtain the necessary information [3]. Agents can work in a Graphical User Interface (GUI) environment that is built for ease of data manipulation by an end-user. The data integration is transparent because the process is embedded in the web page.

**HERMES**

HERMES [15] defines a platform for building mediators, or programs that semantically integrate different and possibly heterogeneous information sources and reasoning systems. Mediators (see the pioneering work by G. Wiederhold, (1992)) typically express methods to resolve conflicts, unify mismatches in measurement units (e.g. convert centimeters to inches) and generate sophisticated conclusions based on information contained in a wide variety of data structures (Lu. Nerode et al., 1996). HERMES mediators employ annotated logic-based rule sets that define precise domain function execution over target data sources. HERMES domain modules encode the actual conduit through which the system accesses native data files.

Subrahmanian, et al. [15] describe the necessity of integrating diverse forms of reasoning within the construction of an agent. The purpose of this method is to extract and produce new information from existing data. This may be necessary to obtain solutions to more complex problems. Construction of an agent as described above requires two important aspects. The first is called domain integration, which is the physical linking of the data sources and reasoning systems. The second is semantic integration, which is the coherent extraction and combination of the information provided by the data and reasoning sources [18]. These agents have reasoning abilities to integrate the heterogeneous data and present a coherent view to satisfy a user query.

The United States Military Academy, Office of Artificial Intelligence, Analysis and Evaluation (OAIAE), in collaboration with the United States Logistics Integration Agency and the University of Maryland (College Park) Department of Electrical Engineering and Computer Science, has prototyped a multimedia tool to visualize the readiness of US Army War Reserve equipment. In contrast to previous efforts, our system allows the readiness to be visualized across several dimensions and to mediate existing heterogeneous legacy data sources. We use a front end Java applet to enable multimedia and multidimensional visualization and to allow a small client to access the HERMES mediation server from any networked location. Current work includes improving a heuristic visual tool to allow humans to generate queries against the data to answer questions that would be very hard for unassisted humans to formulate [17].

**Dynamic Scenario Builder (DSB)**

The Functional Description of the Battlespace (FDB) is a collection of organized, validated, and traceable information that assists developers and military subject matter experts in the development of current and future combat simulations. The FDB contains training objectives as well as the information required to build validated combat scenarios. One example is the JANUS combat simulation. JANUS is an interactive, two-sided, closed, stochastic, ground combat simulation. Currently, a user must obtain verified and valid combat scenarios from a limited number of agencies in order to generate an accepted scenario for a given JANUS application. Given the nature of current warfare, the need exists for the tactical commander to create simulations involving task-organized units within non-traditional theaters of operation. The Dynamic Scenario Builder (DSB) addresses this need.

The United States Military Academy OAIAE in collaboration with the United States Army Simulation, Training, and Instrumentation Command (STRICOM) is currently developing the DSB. The DSB will collect verified and validated data from the FDB and other data sources to build a generic training scenario that will run in a variety of wargaming simulations, including JANUS. The data will be obtained and converted into a form portable across platforms via a web browser. The user of the DSB will access the system through a web browser and employ a Graphical User Interface (GUI) to select the pertinent data necessary for scenario generation. The DSB will enable an authorized user to generate a scenario “on the fly”, resulting in a more efficient use of training time.

To assist in the creation of scenarios, intelligent agent technology will be integrated into the DSB. In addition to providing search and retrieval capabilities to the user, these agents will provide decision support to the user. This intelligence will be embodied in a knowledge base assimilating user requirements, the FDB, libraries of existing scenarios, among other sources.

**5. CONCLUSION**

There are research opportunities into the 21st Century in this area of data management. Researchers are beginning to appreciate the value of intelligent agents for data search, retrieval, and integration. With the plethora of data sources and types now available to users, current approaches to data management are proving inadequate. Standards are being considered in this area, and commercial software agents are being developed for many different purposes, but much more research is required.

The web provides great potential to support information needs of a wide variety of users. Its staggering growth and unstructured nature challenge the efforts of most users efficiently access and assimilate this information, however. Intelligent agents provide a key to solving this problem.
5. REFERENCES


http://www.msci.memphis.edu/∼franklin/AgentProp.htm


