N95- 23675

Software Agents for the Dissemination of Remote Terrestrial Sensing Data

Christopher N. Toomey, Evangelos Simoudis,
Raymond W. Johnson, William S. Mark
Lockheed Artificial Intelligence Center
3251 Hanover St. O/96-20 B254F
Palo Alto, CA 94304
(415) 354-5802 (415) 354-5235 (fax)
toomey@aic.lockheed.com

KEY WORDS AND PHRASES

Software agents, artificial intelligence, remote terrestrial sensing data dissemination, knowledge sharing, national information infrastructure.

INTRODUCTION

Remote terrestrial sensing (RTS) data is constantly being collected from a variety of space-based and earth-based sensors. The collected data, and especially "value-added" analyses of the data, is finding growing application for commercial, government, and scientific purposes. The scale of this data collection and analysis is truly enormous; e.g., by 1995, the amount of data available in just one sector, NASA space science, will reach 5 petabytes. Moreover, the amount of data, and the value of analyzing the data, are expected to increase dramatically as new satellites and sensors become available (e.g., NASA's Earth Observing System satellites). Lockheed and other companies are beginning to provide data and analysis commercially.

The Problem

A critical issue for the exploitation of collected data is the dissemination of data and value-added analyses to a diverse and widely distributed customer base. Customers must be able to use their computational environment (eventually the National Information Infrastructure) to obtain timely and complete information, without having to know the details of where the relevant data resides and how it is accessed. Customers must be able to routinely use standard, widely available (and therefore low cost) analyses, while also being able to readily create on demand highly customized analyses to make crucial decisions.

For example, a company laying an oil pipeline would want processed imagery along the pipeline route (or perhaps along several alternative pipeline routes). This imagery would have certain requirements such as image resolution, spectral band, allowable cloud obscuration, and so on. In order to be useful, this imagery usually has to be processed through various analytical techniques, e.g., registration (to precisely align different images along the pipeline route), elevation determination, feature detection, etc. The purchase of such imagery and processing is often a negotiation process: the information the customer wants may either be unavailable or prohibitively expensive. Customers will usually need to reduce costs by refining their orders based on the availability of standard or pre-existing imagery and analysis products. Thus the oil pipeline company would need active feedback during the order formation process in order to determine how some combination of existing and special order products can meet their requirements.

The diversity of user needs creates a difficult software problem: how can users easily state their needs, while the computational environment assumes the responsibility of finding (or creating) relevant information, and then delivering the results in a form that users understand?

Software Agents

A software agent is a self-contained, active software module that contains an explicit representation of its operational knowledge. This explicit representation allows agents to examine their own capabilities in order to modify their goals to meet changing needs and to take advantage of dynamic opportunities. In addition, the explicit representation allows agents to advertise their capabilities and results to other agents, thereby allowing the collection of agents to reuse each others' work.

A large-scale computational environment for data and analysis dissemination is complex and dynamic, and thus it is unrealistic to expect any human or computer program to acquire and maintain functional knowledge of even a fraction of this environment. It is also unrealistic to think that humans or computer programs will have the expertise to determine the content of an arbitrary database or the requirements and results of a new analysis routine. Therefore, agents must rely on the knowledge that other agents have about their (local) environment. The basic knowledge of a database, analysis routine, or set of user requirements is entered by the humans who define the agent in the first place, such as database administrators, algorithm implementors, or end users. This knowledge is accessed by other agents, which use it to augment and modify their own knowledge of the environment. In this way, the total sum of agent knowledge in the environment is cumulative, taking advantage of new knowledge that is constantly being added to the environment in the form of new agents or human modification of existing agents. At the same time, no agent has to have non-local knowledge about the environment: agents rely on what other agents know, augmenting their own knowledge to improve the efficiency of their ability to interact with other agents (remembering short-cuts, reliable partners, etc.).

TECHNICAL APPROACH

Under funding from NASA's technology commercialization program, we are currently building a "showcase" agent-based RTS data dissemination environment to prove the value of this technology in a real world environment. We are working closely with personnel from Lockheed's Space Systems division and Space Imaging Incorporated subsidiary to ground our effort in reality. The key technologies we are using in this effort are:

- Explicit representation of software capabilities and execution events relevant to multimedia access and analysis.
- Knowledge interchange technology to support the sharing of goals and results among agents.
- Reactive planning technology to enable agents to change their behavior in response to changes in the environment.
- User interface technology to facilitate the specification of agent tasks by a variety of end users.

Explicit Representation of Capabilities and Results

There has been considerable recent research activity directed toward the creation of explicit representations of the capabilities and interests of computer tools. Lockheed has participated in this research, primarily in the representation of engineering knowledge and the capabilities and requirements of engineering tools [1]. We are extending this research to the area of data access and exploitation software, which brings some important new features and challenges. For example, databases are usually structured in terms of abstractions that provide a starting point for the explicit representation; but conventional database abstractions leave out much information that must be supplied in the knowledge base.

Knowledge Interchange Technology for Agent Interaction

Government agencies, telephone and other communication companies are developing the network infrastructure that is making efficient large-scale dissemination of data and derivative products cost effective. A key part of this infrastructure is knowledge interchange technology that allows distributed heterogeneous software components to take full advantage of the communication enabled by the new bitways. The knowledge-sharing infrastructure includes a common knowledge representation language, domain ontologies, standard agent/tool interaction protocols, and a facilitation services such as consumer/producer matchmaking [1,2]. Database and analysis tools "plug in" to this infrastructure via wrappers. Wrappers provide an interface that translates between internal tool representations and the shared language and protocols of the infrastructure. Lockheed is a major participant in the creation of the knowledge-sharing infrastructure and wrapper technology. This technology forms the foundation of our agent-based data dissemination environment.

Reactive Planning for Dynamic Behavior Modification

A key tenet of our approach is that agents must be able to examine the capabilities and results of other agents to achieve their goals. In order to actually use this knowledge, agents must act opportunistically, modifying their goals to make use of the partial results and ongoing pursuits of other agents. For example, agents must be able to dynamically reformulate their action plan if they receive a message that another agent has already achieved one of the intended results of their actions. Reactive planning technology enables agents to dynamically change their plans and behavior in response to relevant changes in their environment [3].

User Interface Technology Facilitating Agent Task Specification

We are utilizing advanced user interface technology to ensure that all types of end users will be capable of using our agent-based RTS data dissemination system. Our interface technology hides the complexity of the underlying system by allowing users to interact with the system via high-level, forms-based graphical user interfaces that use standard terminology from the remote sensing domain.

STATUS

We are about halfway through our initial contract with NASA to demonstrate the use of software agent technology in addressing the RTS data dissemination problem.

Progress to Date

To date we have completed a working agent-based prototype for Space Imaging's customer service center (CSC) and representative data sources that it will access. The CSC is the software interface between customers and remote terrestrial sensing products (data and analyses that meet the customer's needs).

Figure 1 illustrates the architecture of our current customer service center prototype. The system demonstrates access to a variety of data sources: archives of images from specific satellites (Landsat, Spot, and Lockheed's own Space Imaging Incorporated (SII) satellite); a database of low resolution preview images, or "chips"; and the SII satellite itself, which can be tasked to produce new images, and thus act as an active data source. Reflective of the real world environment, these data sources are distributed and heterogeneous (implemented using different database management systems and different data representations).

The user interacts with the CSC system via a high-level graphical user interface (GUI). The GUI includes several features to simplify the order specification process. First, it allows the user to specify the desired imagery's geographic region location by drawing it directly on a scalable world map. Second, it allows the user to specify constraints on other image attributes (such as resolution and image acquisition date) via forms-based templates that use generic RTS domain terms and values rather than database-specific ones. Third, the system recommends settings for different attributes based on the application domain selected by the user (e.g., one meter resolution imagery for property assessment applications).

The central element of the system is the Data Broker agent, which serves as the intermediary between the customer and the data sources. The Data Broker receives formal descriptions of the desired imagery characterized by location, resolution, acquisition date, etc. from the GUI. It is responsible for matching data requests to a set of specific data sources capable of providing such data. The Data Broker is aware of the capabilities and input requirements of data sources because they have been advertised. Data sources come on line when their wrapper agents advertise their capabilities to the other agents in the environment, including the Data Broker. The Data Broker is thus able to transform incoming data requests into "targeted data requests" based on the known capabilities and requirements of all available data sources.

Wrapper agents for the individual data sources receive these targeted data requests, and are responsible for returning metadata for each of the images they have meeting the user's requirements. To do so, a wrapper translates the request from the common interagent language into the wrapped data source's query language, queries the data source, and translates the results back into the interagent language.

Lastly, the Data Broker is responsible for collecting and pruning the wrappers' results in order to create a coherent composite result set. Pruning is necessary when the different data sources provide overlapping results. It can be a task of considerable sophistication, since it can require making tradeoffs on different data characteristics (which is better, less cloud obscuration or higher resolution?). Currently, the Data Broker supports only a single pruning option: to remove older images in the result set that are completely overlapped by newer ones.

Future Work

The CSC prototype shown in Figure 1 is implemented and is end-to-end operational. However, only the Spot and SII Archives data sources have been wrapped and put on-line to date. In the remainder of this year we will be wrapping the other data sources, including the SII satellite tasking module, which will require reactivity to collection scheduling changes induced by bad weather, crisis tasking requests, and order tasking conflicts. We will also be illustrating result sharing among agents, such as between multiple Data Broker agents, and the use and management of a dynamic collection of persistent agents representing customer orders [4].

REFERENCES

[1] McGuire, J. G. et. al., 1993. SHADE: Technology for Knowledge-Based Collaborative Engineering. *Concurrent Engineering: Research and Applications* 1(3).

[2] Simoudis, E., 1992. Integrating Distributed Expertise. *International Journal of Intelligent and Cooperative Information Systems* 1(3-4).

[3] Kaelbling, L.P.; and Rosenschein, S.J., 1990. Action and Planning in Embedded Agents. *Robotics and Autonomous Systems* 6(1).

[4] Toomey, C.N.; and Johnson, R.W., 1994. Software Agents for Automating Multiple-Tool Tasks. In *Working Notes of the 1994 AAAI Spring Symposium on Software Agents.* Palo Alto, Calif.: American Assoc. for Artificial Intelligence.

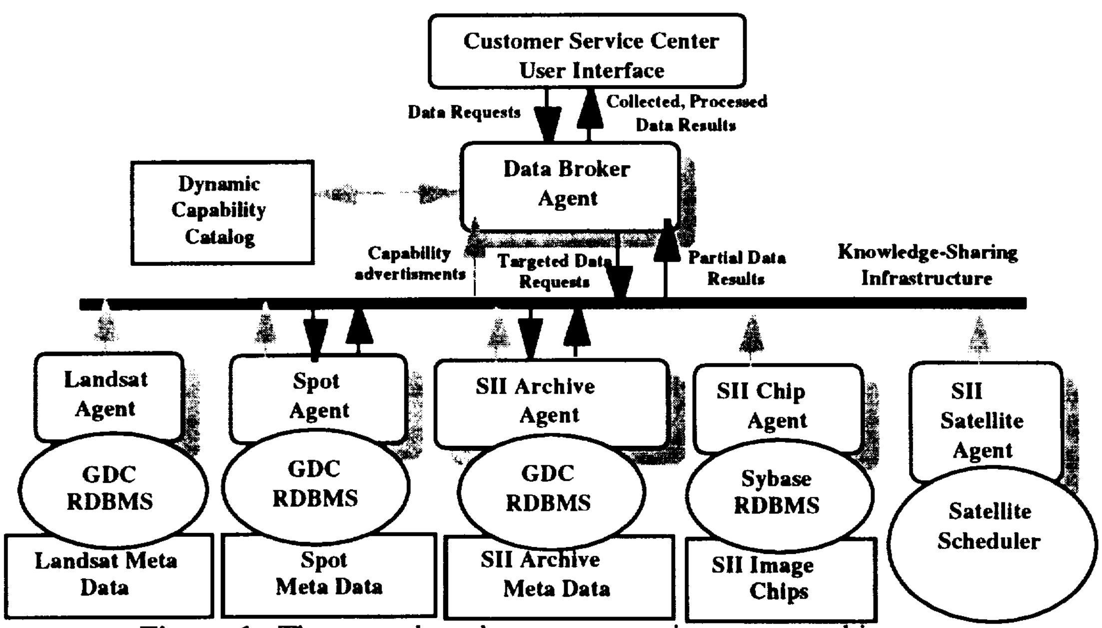

Figure 1. The agent-based customer service center architecture.