MOBILE AGENT APPLICATIONS

Rosane Maria Martins, Magali Ribeiro Chaves, Luci Pirmez and Luiz F. Rust
NCE/UFRJ - Núcleo de Computação Eletrônica
Universidade Federal do Rio de Janeiro
Tel.: +55 21 598-3159 - Caixa Postal: 2324 - Rio de Janeiro RJ Brasil
{rosanemartins, magalichaves}@uol.com.br, {luci,rust}@nce.ufrj.br

Abstract. *Automatic, autonomous browsing has an increasingly important task in information discovery and assisted browsing on the Internet. Where users could once keep up to date with information of interest on the Internet, the recursive growth of the network has made this process increasingly time consuming and less rewarding. We present two possible solutions to this problem: Data Agents and CollaborAgents which were developed with IBM's Aglet Workbench – a particular implementation of mobile agents. This paper also surveys the agent technology and discusses the agent building package used to develop both mentioned applications. Finally, it concludes that the future of local interaction, reduced network loading, server flexibility and application autonomy which are supported by mobile agent technology all help to provide a level agility above distributed problem solving.*

1 Introduction

Internet based electronic commerce of today has not made shopping that much easier for the consumer. In fact, it has maybe even made it worse. We have more information about products than ever, more stores to visit. The difference between the electronic marketplace and the real-world marketplace is that you visit the store electronically instead of physically.

So, finding and combining the relevant information/service is becoming a critical task. There is a need for facilities that perform these integrating tasks and thus overcome problems such as distribution and heterogeneity. These facilities are often referred to as integrated systems. In an integrated system, the user is not exactly aware of which and how many information sources that are used, neither does he know how they are used. The user is provided with the vision that only one information source exists.

To address this problem, several exciting new technologies have been developed. Several groups of Artificial Intelligence (AI) researchers are already actively involved in trying to design Internet assistants that will make easier the filtering or retrieving information from the network and the virtual purchases.

Each intelligent assistant is composed by autonomous agents and is based on AI and Distributed Artificial Intelligence (DAI) concepts.

DAI is concerned with all forms of social activity in systems composed of multiple computational agents[1]. An important form of interaction in such systems is cooperative problem solving, which occurs when a group of logically decentralized agents choose to work together to achieve a common goal.

The "agent" term is largely used in different areas, such as Distributed Systems or Software Engineering, for this reason, there are almost many definitions for it. However, agent systems present key characteristics which differ them from other softwares.

Agents are autonomous, persistent (software) components that perceive, reason, communicate and act in someone's favour, influencing its environment. This environment presents many agents which will interact. This interaction is the Multiagent Systems' principal element .

This paper contributes with two applications that emphasize the mobile agents technology as an significant and revolutionary paradigm for distributed problem solving: *Data Agents*

and *CollaborAgents*. Both were implemented using IBM's mobile agent framework known as Aglets Software Development Kit (ASDK).
Data Agents is a mobile and intelligent multiagent auxiliary prototype for the retrieval of distributed structured information in a scenario of several "on-line" bookstores. The proposed system is based on a group of agents trying, simultaneously, to find products of users interest in several virtual places known by them, presenting the results in an homogeneous way.
CollaborAgents is a prototype where users create autonomous agents to buy and sell goods in a context of the automobile components distribution problem. A car assembler looks for suitable supplies to attend the subsidiaries' orders. This commercial transaction is done through the Internet.
This paper is organized as follows. An overview of agent concept and detailed descriptions of a mobile agent environment known as Aglets WorkBench are given in section 2. Section 3 describes the overall architecture of two applications: Data Agents and Collaboragents. Finally, section 4 presents the conclusion and future work.

2 Mobile Agents

Mobile Agents are processes dispatched from a source computer to accomplish a specified task[3,4]. Each mobile agent is a computation along its own data and execution state. In this way, the mobile agent paradigm extends the RCP communication mechanism according to which a message sent by a client is just a procedure call. After its submission, the mobile agent proceeds autonomously and independently of the sending client. When the agent reaches a server, it is delivered to an agent execution environment. Then, if the agent possesses necessary authentication credentials, its executable parts are started. To accomplish its task, the mobile agent can transport itself to another server, spawn new agents, and interact with other agents. Upon completion, the mobile agent delivers the results to the sending client or to another server.
In order for these agents to exist within a system or to themselves form a system they require a framework for implementation and execution. This is known as the agent environment.

2.1 Aglet Technology: A Java Based Agent Execution Environment

Developed by the IBM Japan research group, this package is a framework for programming mobile network agents in Java. We can use a few expressions to describe an Aglet[5]: written in pure Java, light-weight object migration , built with persistent support, event-driven. It is easy to understand why JAVA is necessary for WAN application's existence in today's heterogeneous networking environment. Besides providing platform independence, JAVA also provides sandbox security to protect host against malicious attacks from alien applications.
Unlike an applet´s short and boring period of execution, an aglet can exist and execute tasks forever. One of the main differences between an aglet and the simple mobile code of Java applets, is the itinerary that is carried along with the aglet. By having a travel plan, aglets are capable of roaming the Internet collecting information from many places. The itinerary can change dynamically giving the aglet the sense of self-governing and the look of an intelligent agent (that of course is in the hands of the programmer).
An aglet can be dispatched to any remote host that supports the Java Virtual Machine. This requires from the remote host to have preinstalled Tahiti, a tiny aglet server program implemented in Java and provided by the Aglet Framework. A running Tahiti server listens

to the host´s ports for incoming aglets, captures them, and provides them with an aglet context (i.e., an agent execution environment) in which they can run their code from the state that it was halted before they were dispatched. Within its context, an aglet can communicate with other aglets, collect local information and when convenient halt its execution and be dispatched to another host. An aglet can also be cloned or disposed.

3 Applications

Two applications based on agent technologies were implemented in this paper: Data Agents and CollaborAgents.

3.1 Data Agents – A Group of Mobile Agents for E-Commerce

The agents for electronic commerce considered in this context are those that somehow help the users to shop over the Internet. This type of agent, called shopping agent, may carry out several tasks, such as: to help the user decide what product should be purchased; to make suggestions based on its knowledge of its owner; to find out new things, discounts and special prices; to find stores that sell the desired product or service, among other things.
In order to show the feasibility of the search process for structured and distributed information through the mobile agents technology, this paper proposes the development of a multiagent, mobile and intelligent system called "*Data Agents*" in the context of several "on line" bookstores. The goal is to accelerate the retrieval of distributed structured information. This is achieved by improving the phase of the process of data selection, in which the agents run parallel among the servers related to them and at the end returning with all the information requested by the user, without the need to make a call to each one of the servers separately. The information obtained is then presented in a uniform and organized way. Using the information thus presented by the system, it is much easier for the user to choose a product with the most satisfactory characteristics.
Among the possible functions described above, the Data Agents Agent is intended to help find the stores that sell the desired product and to list the prices of the products found.
The operation of the prototype to achieve this objective is the following:

- user selects the specific product and the desired characteristics of that product (these characteristics will be the restrictions for the search);
- the purchase agent searches for products with the desired characteristics among products of that type;
- as a result of the search, Data Agents sends an e-mail or shows a screen to the user with a list of products, their respective prices and where they can be found.

3.1.1 Proposed Architecture

The following architecture is proposed to enable the Data Agents system to have the functionality mentioned above and, in the future, to be applied to many products and stores.
The system presented comprises the following components: Interface Module, Control Module and Purchase Agents. What follows is a detailed list of the system components:
Interface Module: this is the component through which the user contacts the system and places his order. This module is also responsible for presenting the result obtained by the group of agents to the user.

“Title”, “author” ,“price range” and "type of itinerary" are the information that the user must provide to the Interface Module so that it may request the Control Module to create and dispatch the purchase agents according to the restrictions imposed by the user.
There are three possibilities of choices for itineraries:
a. *one agent for each server:* According to the quantity of servers registered in the system, one agent is created for each server and dispatched to do its task. When each agent arrives at its destiny, it does its search, send the result as a message to Control Module and "dies".
b. *only one agent that visits all the servers:* It is created only one agent that has in its travel plan the addresses of all servers. It will go to all servers, one by one, do the search, send the result as message to Control Module and "dies" at last visited server.
c. *one agent that goes through the servers until to find the first occurrence*: It is created only one agent that contains in its travel plan the addresses of all servers. But it will travel to next server only if doesn't find any book at former server, that is, the agent travels until to find the first occurrence that satisfies the order user.
As soon as the result manager (a component of the control module) compiles all answers received, it sends these answers to the interface module so that they are delivered to the user: on the screen or via e-mail.
Control Module: This module is responsible for the creation and release of purchase agents to begin the search requested by the buyer. This module also aggregates the results found by the different agents. There is a control module for each type of product available in the system, e.g. a control module for books and a different one for CDs.
After receiving the user’s requirements from the interface module, the Control Module creates the agents according to such requirements and sends them to the addresses available at a Storage Structure.
Storage Structure is a hash structure that contains the addresses of the various stores associated with the system. There is a storage structure for each type of product researched by the system.
When the Control Module receives a request to send an agent, the latter is created on the “aglet” layer according to user's requirements and travels through the runtime layer, which converts the agent into an array of bytes and such array, on its turn, passes on to the ATP layer – Agent Transfer Protocol, to be sent to its destination. This protocol, then, builds a “bit stream” that contains both general information, such as the system name, and its identification, such as the “byte array” resulting from the runtime layer.
Upon returning to the server with the information from its search, each purchase agent sends its contents to the Result Manager (Control Module), so that the Results Manager may aggregate all answers obtained and send them to the interface module.
Purchase Agents : Make contact with the stores by accessing their databases, place the order and interpret the answers generated, converting them into a format that is understood by the control module. Before proceeding to their destination, the agents are coded in bit stream: the first segments are general information, such as the agent’s identification, and the last segment is the byte array, the agent per se: code and state. The goal of the agents is to check the information found at their destination address, selecting only the information considered relevant and recommended according to the pre-determined rules. Such information shall represent the basis of rules to be used by the agent to make appropriate decisions in the process of evaluation of the items found.
With this architecture, the extension of this system to deal with new products and new stores is simple, although it is necessary to build a control module for each new product.

3.2 CollaborAgents – Solution for a Logistic Problem Applying Multiagent Systems

The Internet has been extensively explored as an environment which brings great ease to integrate clients and suppliers willing to negotiate products and services. Under that light, an area that deserves special attention is the automatic negotiation between clients and suppliers.
The negotiation model presented consists of a system of agents that acts in the process of integration among clients, represented by the subsidiaries of an automaker, and a network of suppliers. This system contributes to the mastering of electronic commerce, since some client agents interact with supplier agents trying to find products and buy products that meet their needs.
The subsidiary which wishes to purchase a product may ask the client agent to initiate the negotiation with a network of supplier agents remotely distributed on the Internet. After a number of interactions with other agents the client agent returns to its original computer and shows the result of its negotiations, i.e. a list of suppliers that best fit its needs in terms of price, freight cost and quantity of product.
As supplier/client agents are created, they are sent from their original computer to an agency, where they will communicate to achieve their overall goal. The Agency, or Meeting Place, is a host computer, where the agents do business. Each agency represents a certain region. In that respect, the client agent will run the network searching for possible suppliers to its demand, and will prioritize agencies located in regions closest to the location of the subsidiary represented by the agent , in order to reduce freight costs.
This section of the paper will try to describe the components forming the model developed and analyze its operation.

3.2.1 Architecture of Model

The architecture of the model is characterized by the exchange of messages among three categories of agents: facilitator agent, supplier agent and client agent.
Facilitator Agent: It is responsible for managing the negotiation between client agents and agency suppliers. The facilitator agent works as an intermediary for such agents. The facilitator records all suppliers with their respective offers and indicates to the client agent the best supplier to establish the negotiation process with. The facilitator agent has an optimization module to carry out that job. That optimization module inquires each supplier able to meet the subsidiary's demand through the "SearchOffer" message and decides which is the most favorable candidate based on which supplier made the best offer (lowest cost).
Supplier Agent: It represents the interests of the supplier. Interests of the supplier means the offer of parts and their respective costs (unit price and freight).
A List of Offer of Material Form is opened when the supplier agent is created. The description of the part, the quantity available for sale and unit price will be typed into the List of Offer of Material.
After confirmation of the information typed in the list of offer of material, the supplier agent will be sent to an agency. At the host computer the agent will make the first contact with the facilitator agent, subsequently requesting to be listed in the database by using the "Register" message.
The supplier agent is responsible for calculating the total cost of the material requested by a subsidiary (price of merchandise, including freight). It is noteworthy that the freight legislation is too wide and this paper is not intended to study all rules and effectiveness in each State. Suppliers hire the service of a shipper, which calculates that cost. This information will be sent to the supplier and will become part of the context of its agent.

This case highlights the main advantage of the object-oriented programming: polymorphism. Each class of supplier has a different internal policy to carry out the same method (calculation of freight) and the facilitator agent is not responsible for knowing the procedures of all suppliers. When the facilitator agent inquires the supplier agent in relation to the cost of the material requested by the subsidiary, that agent is not interested in the internal details of that calculation, only in the result.

When the entire inventory of the supplier agent has been negotiated, the agent will send a "Unregister" message, requesting to leave the list kept by the facilitator agent. After its exclusion from the database, the supplier will return to its original computer and will show to the user the result of its negotiation with the different client agents that contacted the supplier agent.

Client Agent: It represents the interests of a subsidiary. This means the demand for parts that a subsidiary is willing to buy.

A List of Request of Material form is opened when the subsidiary agent is created. That form will contain the subsidiary's identification, description and location entered by the user, as well as the description and quantity of the material requested, later on confirming the information.

After that information is confirmed, the client agent will migrate to an agency searching for the best supplier to meet its demand. After arriving at the agency, the agent will communicate with the facilitator, which will indicate the context of the best suitable supplier. Upon obtaining the answer, the client agent will initiate the negotiation process through the "Negotiate" message. If the client agent does not find the desired supplier, it will go to other agencies searching for new proposals to meet its demand. After making all negotiations, the client agent will return to its original computer and show the result of its interaction with other agents.

4 Discussions and Future Work

We have built two simple prototypes to test the basic concepts and feasibility and conducted some simple experiments.

Nowadays, the Data Agents system presents a simplified configuration, allowing just one kind of product to be investigated. The selected product to this stage was "book". The purchaser agent is created at the user machine and migrates to the target server. This target server is an aglet server where it may be found the offered service by the supplier.

The results obtained in this initial phase were considered acceptable, due to the information filtering be executed on the server where the resources are located. This circumstance leads to a significant reduced network loading.

The possibility of existence of more than one kind of product to be investigated is a task to be implemented in another phase of this project. Another improvement is an implementation of an intelligent module that permits the Data Agents system, besides of the conventional search, makes suggestions to the user.

Concerning the CollaborAgents system, a logistic model was implemented to solve the issue of automotive parts distribution. In this case, each subsidiary and supplier creates an agent which shares information seeking for a global solution.

From this experimentation, it was obtained good results, once the implementation of mobile agents avoided the overload on the web. In a next stage, it will be developed a sophisticate negotiation scheme for buyers and sellers. We are studying two models : first, involving "price-raise" and decay functions similar to Kasbah's technique [2] and the second, based on computational intelligence, where the agents will be able to make proposals through the experience that they acquired a long their life cycle.

Future works is focused on making smarter agents which are directable at a more natural level for users. Though we have only just scratched the surface in terms of making a truly useful system, we are excited about these works and think they have the chance to essentially change the way people acquire goods and services in the not-too-distant future.

References

1. Bond, A.H.; Gasser , Les.(Eds) Readings in Distributed Artificial Intelligence. San Mateo, California: Morgan Kaufmann, 1988.
2. Chavez, A. and Maes, P.: An Agent Marketplace for Buying and Selling Goods, Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, UK, April 1996.
3. Chess, D.; et al. Itinerant Agents for Mobile Computing. Journal IEEE Personal Communications, Vol.2, Nº 5, October, 1993.
4. General Magic Inc. Mobile Agents. http://www.genmagic.com
5. IBM Japan Research Group. Aglets Workbench. http://aglets.trl.ibm.co.jp