EXPERIMENTS IN THE MOBILE AGENT TECHNOLOGY

Rosane Maria Martins, Magali Ribeiro Chaves, Luci Pirmez e Luiz Fernando Rust
NCE/UFRJ - Núcleo de Computação Eletrônica
Universidade Federal do Rio de Janeiro
Tel.: 021 598-3159 - Caixa Postal: 2324 - Rio de Janeiro RJ Brasil
{rosanemartins, magalichaves}@uol.com.br, {luci, rust}@nce.ufrj.br

The beginning of Internet as commercial usage with its easy graphic interface originated an enormous increase of users' quantity as well as a great variety of available services during the last years. Nowadays the majority of private and public institutions are prepared to provide any kind of information or service through the Internet.

The paper describes agent technology and a particular implementation of mobile agents, IBM's Aglet Workbench. This is followed by the description of two applications of Aglet system: Bon Marché and Logistics Agents.

The paper 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

The trend toward computer-based tools for many aspects of commerce has led to a rapid increase in distributed virtual workgroups, such as multi-vendor design teams and virtual corporations. In addition, the advent of the Internet, personal computer networks, and interactive television networks has led to an explosion of information available on-line from thousands of new sources. This phenomena offers great promise for obtaining and sharing diverse information conveniently, but they also present a serious challenge. The sheer multitude, diversity, and dynamic nature of on-line information source makes finding and accessing any specific piece of information extremely difficult.

To address this problem, several exciting new technologies have been developed. Many groups of Artificial Intelligence (AI) researches 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 the 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 others 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 between themselves. This interaction is the Multiagent Systems' principal element .

This paper contributes with two applications which emphasizes the mobile agents technology as an significant and revolutionary paradigm for distributed problem solving : *Bon Marché* and *Logistic Agents*. Both were implemented using the IBM's mobile agent framework known as Aglets Software Development Kit (ASDK).

Bon Marché is a mobile and intelligent auxiliary multiagent software to the electronic market in order to make easier the Internet's purchases. The proposed system is based on a group of agents trying to find simultaneously the users interest products in several virtual places known by them, presenting the results in an homogeneous way.

Logistic Agents are a prototype designed to help 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 by Aglets WorkBench are given in section 2. Section 3 describes the overall architecture of two applications: Bon Marché and Logistic Agents. Finally, section 4 gives the conclusion and future work.

2 Agent Technology

2.1 Agent Concepts

'The term 'agent' has been picked up, widely appropriated, and in many cases misappropriated, by technical publications, lay publications, and many researchers in computer science.'[3]

The question of what actually constitutes an agent, and how they differ from a normal program, has been heavily debated for several years now[8]. Agents can be loosely defined as *'software that assist people on their behalf, …and are delegated to perform task(s), and given constraints under which they can operate'* [7]. However, agents come in a myriad of different types, usually depending on the nature of their environment. What has been needed is a classification scheme to distinguish between different types of agents. This paper does not propose a further definition of what an agent is, but will adopt the Franklin and Graesser [4] classification scheme and categorise the agents dealt with here as goal-oriented, communicative, mobile agents.

Franklin and Graesser definitions are as follows :

1.*goal oriented agents* - agents that do no simply act in response to the environment;

2.*communicative agents* - those able to communicate with other agents;

3.*mobile agents* - those able to transport themselves from one host to another.

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.

2.2 Agent Environments

There are a large number of agent building packages on the market that allow users to attempt to build and manage their own agents and agent systems. First, and probably foremost is the Tabriz AgentWare package from General Magic[6], which executes and manages agent-based applications on servers, and Tabriz Agent Tools for creating agent applications deployable on Web sites.

General Magic Inc. invented the mobile agent and created Telescript, the first commercial mobile agent system. Based on a proprietary language and network architecture, Telescript had a short life. In response to the popularity of the Internet and later the steamroller success of the Java language, General Magic decided to reimplement the mobile agent paradigm in its Java-based-Odyssey. This system effectively implements the Telescript concepts in the shape of Java classes. The result is a Java class library that enables developers to create their own mobile agent applications.

ObjectSpace's Voyager is a platform for agent-enhanced distributed computing in Java. While Voyager provides an extensive set of object messaging capabilities, it also allows objects to move as agents in the network. You can say that Voyager combines the properties of a Java-based object request broker with those of a mobile agent system. In this way, Voyager allows Java programmers to create network applications using both traditional and agent-enhanced distributed programming techniques.

The Java Agent Template (JAT) architecture provides a set of packages that facilitate building multi-agent systems using the Java programming language. The recent release of the JATLite[5] package has gone some way to simplifying the process, by providing different ‘layers’ from which to abstract, thus offering a flexible base on which to build.

IBM has developed two packages – the Agent Building Environment (ABE), and the Aglets Workbench. The ABE is a toolkit for software developers that makes it easy to build an application based on intelligent agents or to add agents to an existing one. Once again the package includes a number of pre-built components from which it is possible to add agent technology to applications.

The Aglets Workbench, developed at IBM's research labs in Japan, is aimed at producing stand-alone mobile agents. The complete package offers a graphical environment for building mobile agent applications in Java, an agent server, and the specification for an Agent Transfer Protocol (ATP). The experimental work discussed later has been achieved through use of the Aglets Workbench.

2.2.1 Aglet Concepts and Architecture

Aglet is a java-based internet agent. We can use a few phrases to characterize an Aglet: 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.

Aglet is different from distributed object model in that computation itself is transmitted, while for distributed object models, we actually transmit the requests for remote methods. Aglet adopted another approach, it uses a technique called serialization to transmit data on the heap and migrate the interpretable byte-code.Aglet has well-defined entry point for itself to re-start computation.

Aglet also comes with support for persistence. By calling appropriate base-class functions, we can temporarily store aglets in secondary storage and later activate it.

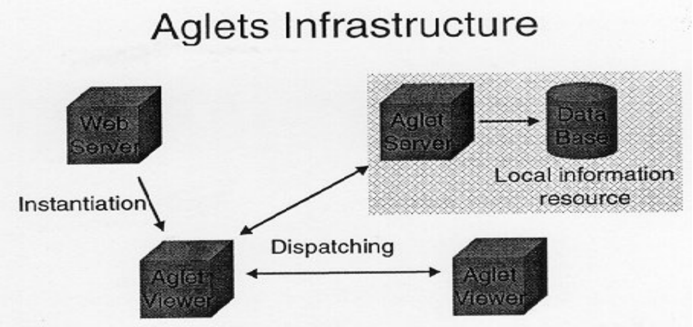


Figure 1 - Aglets Infrastructure

The big figure in an Aglet world has the following components: an aglet viewer such as tahiti, an aglet server and finally aglets themselves. Aglet viewer is in many sense an applet viewer. Besides that, It further allows you to create, retract, activate, deactivate and dispatch aglets. It is a client-side control center.

Aglet server are powerful machines that can host large number of aglets and typically with large amount of data or computing resources. Aglets live in the context of hosts. Hosts enforce security policy by configuring its security manager. When dispatched, each aglet will carry an itinary. It will follow the itinary to choose its own routing. Aglets in their lifetime will visit several aglet hosts, perform computation tasks at host machines and finally carries the result back.

3 - Applications

Two applications based on agent technologies were implemented in this paper: Bon Marché and Logistics Agents.

3.1 - System Description: Bon Marché – A Group of Mobile Agents for Commerce in the Internet

The agents for electronic commerce considered in this context are agents that somehow help the users to shop in 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 novelties, 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 process of search of structured and distributed information through the technology of mobile agents, this paper proposes the development of a multiagent, mobile and intelligent system called "*Bon Marché*" 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 all the information requested by the user, without the need to make a call to each one of the servers separately. The information obtained are then presented in a uniform and organized form. Using the information thus presented by the system, it is much easier for the user to choose a product with the most satisfactory characteristics.

Amongst the possible functions described above, the Bon Marché 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 such objective is the following:

- user selects the specific type of product and, if desired, the desired characteristics for that product (these characteristics will be the restrictions to the search);
- the purchase agent searches for products with the desired characteristics among products of that type;
- as a result of the search, Bon Marché 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 Bon Marché Agent to have the functionality mentioned above and to, in the future, be applied to many products and stores:

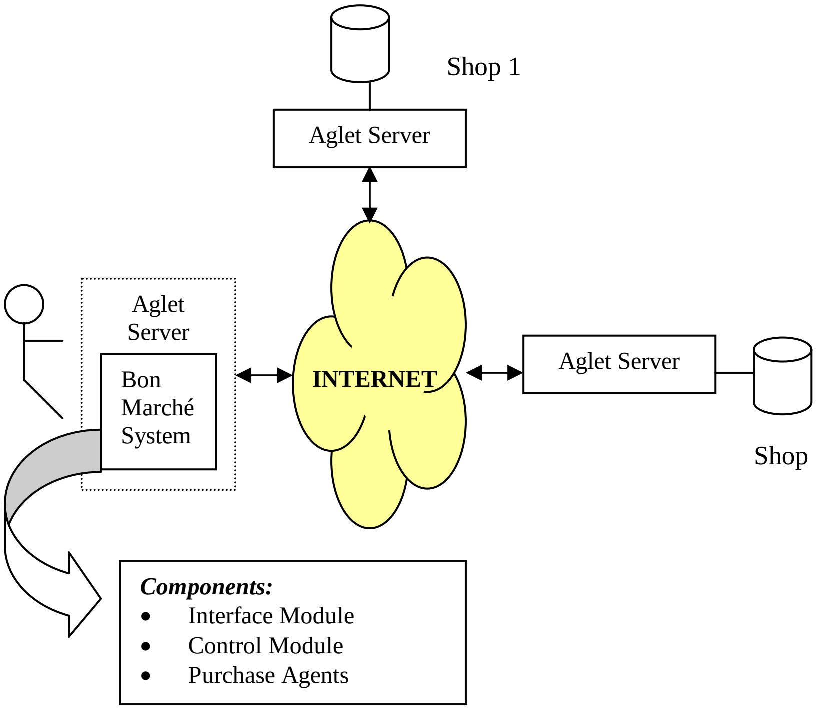
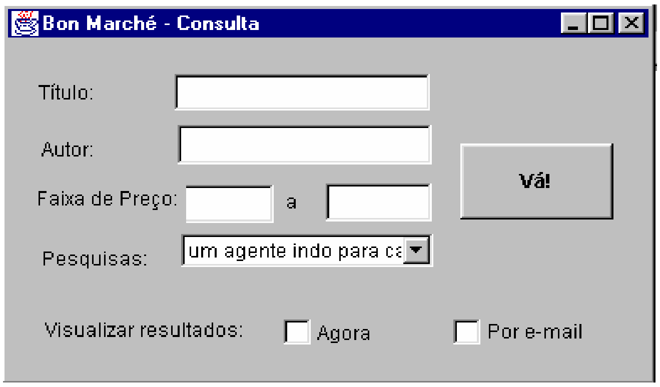


Figure 2 – Overall Architecture of the Bon Marché Purchase Agent

As seen in figure 2, the system presented comprises the following components: Interface Module, Control Module, Purchase Agent and Storage Structure. 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.



Bon Marché - Consulta

Título:

Autor:

Faixa de Preço: a

Pesquisas: um agente indo para ca

Visualizar resultados: □ Agora □ Por e-mail

Figure 3 - Interface Module

“Title”, “author” and “price range” are information that the user must provide to the Interface Module so that the Interface Module may request the Control Module to create and send the purchase agents according to the restrictions imposed by the user.

As soon as the result management module (a component of the control module) compile all answers received, it send such answers to the interface module so that the answers 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 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.

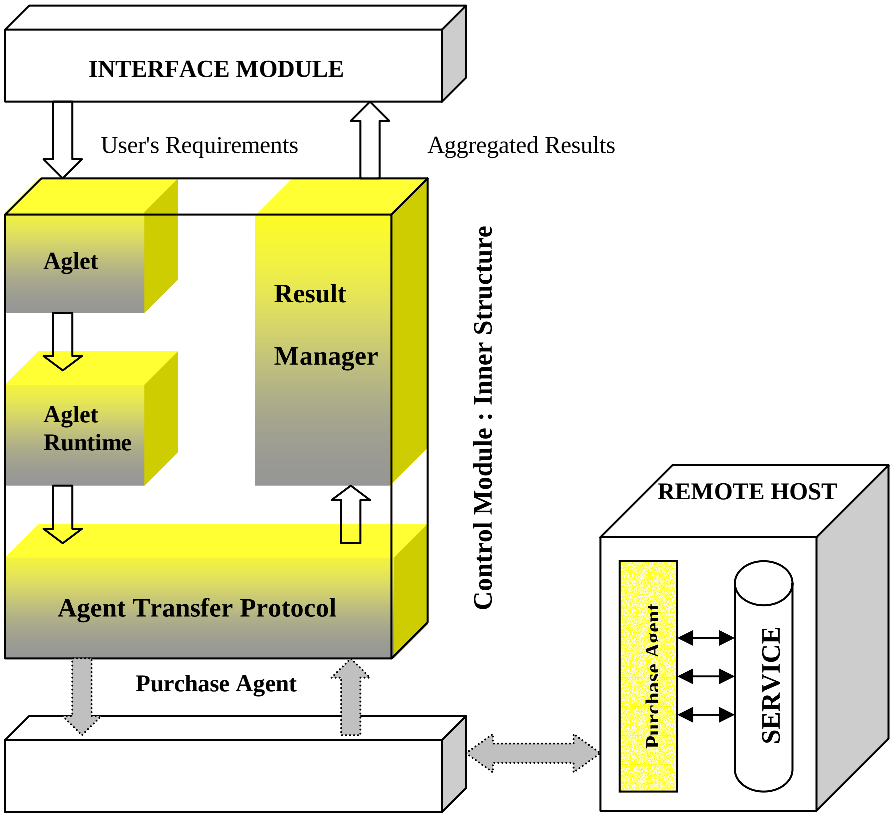


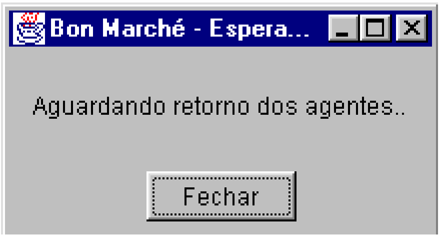
Figure 4 - Control Module - Inner Structure

This module controls the operation of the system as a whole, managing all information obtained from the different agents. This module is directly responsible for the creation, send and receipt of the agents.

As can be seen in figure 4, 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



Aguardando retorno dos agentes..

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.

Figure 5 - Interface of Control Module

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.

The rationale logic to be used will be the Forward Chaining. According to the value of the variable statusBook, the agent will decide whether to send the information found to the result manager.

With this architecture, the extension of the purchase agents 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 – System Description: Logistics Agents – 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, as shown in Figure 6. 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.

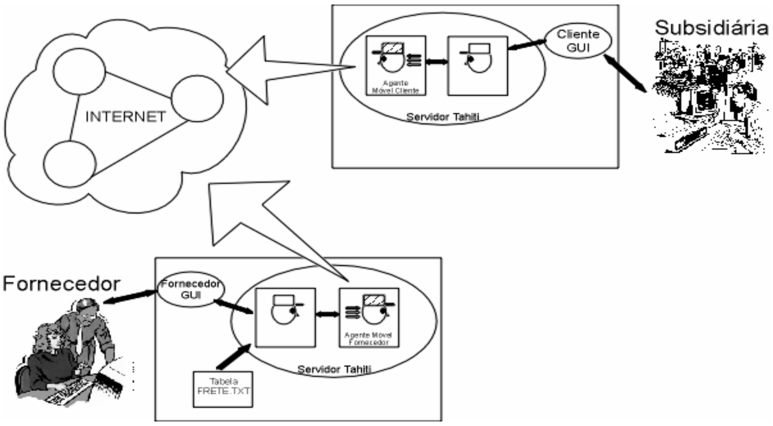


Figure 6 - Agents are sent to Meeting Places

This section of the paper will try to describe the components forming the model developed and analyze its operation.

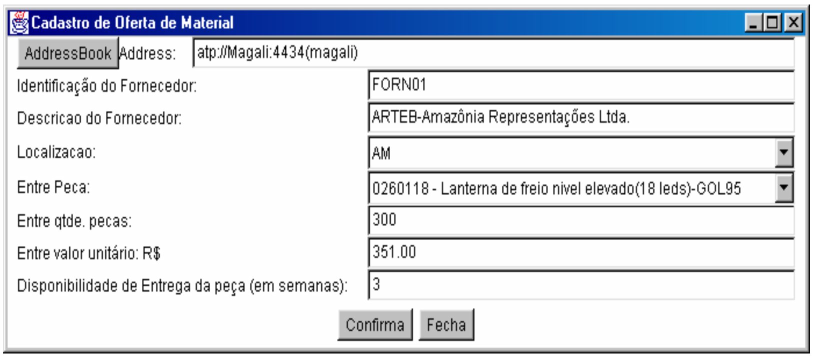
3.2.1 Architecture of Model

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:** This Agent 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 shown in Figure 8 and decides which is the most favorable candidate based on which supplier made the best offer (lowest cost).

- **Supplier Agent:** This Agent 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, shown in Figure 7, 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.



Cadastro de Oferta de Material

Address:	atp://Magali:4434(magali)
Identificação do Fornecedor:	FORN01
Descricao do Fornecedor:	ARTEB-Amazônia Representações Ltda.
Localizacao:	AM
Entre Peca:	0260118 - Lanterna de freio nivel elevado(18 leds)-GOL95
Entre qtde. pecas:	300
Entre valor unitário: R$	351.00
Disponibilidade de Entrega da peça (em semanas):	3

Figure 7 List of Offer of Material Form

After confirmation of the information typed in the list of offer of material, the supplier agent will be sent to an agency.

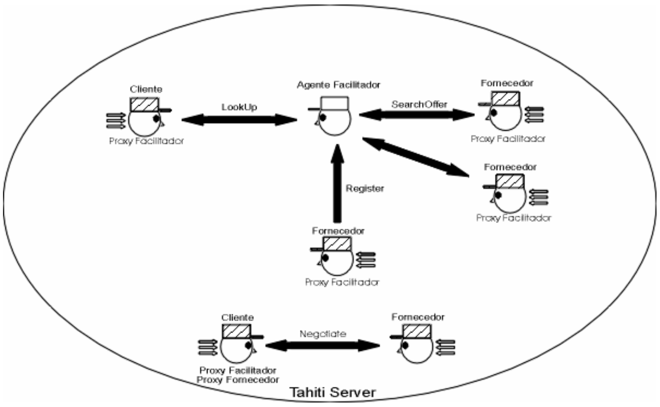


Figure 8 A Meeting Place : Inner Structure

At the host computer the agent will make the first contact with the facilitator agent, subsequently requesting to be listed in the database represented by Figure 9 by using the “Register” message.

COD-FORN	**COD-PEÇA**	**QTDE_OFER**
Forn001	PC001	2000
Forn001	PC002	1000
Forn002	PC001	1500

Where :
COD_FORN = Supplier’s code
COD_PECA = material’s code
QTDE_OFER = Quantity available to sale

*The principal key for this database is formed by COD_FORN + COD_PEÇA

Figure 9 Supplier Database

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. The example shown in Figure 10 shows the table of costs used by a supplier agent. Notice that the cost of freight varies according to the distance. Transportation of a unit of part 0260118 from Manhumirim, MG (supplier's location) to Vila Velha, ES costs R$116,00 (for conventional shipping) or R$200 (express shipping). If the same part is shipped to Fortaleza, CE, freight would cost R$150,00 or R$256,00, depending on the type of shipping selected.

CD_PC	**DEST**	**CST_PC (R$)**	**CFC (R$)**	**CFE (R$)**	**CFUC (R$)**	**CFUE (R$)**	**SEGURO (%)**
0260118	**ES**	**500,00**	**116,00**	**200,00**	**11,66**	**22,00**	**0,17**
0260118	**CE**	**500,00**	**150,00**	**256,00**	**15,00**	**25,60**	**0,22**

Where :

CD_PC = material's code;
DEST = Destination. The client's city;
CST_PC = unit price of the part;
CFC = conventional shipping cost (for a unit of part);
CFUC = conventional shipping cost (when there is more than one unit of part to be shipped);
CFE = express shipping cost (for a unit of part);
CFUE = express shipping cost (when there is more than one unit of part to be shipped);
INSURANCE = cargo insurance;
Qtde_Req = unit quantity required by the client (subsidiary)

Figure 10. Table of costs used by a supplier agent

The calculation of freight at both categories is represented by formulae (ii) or (iii) of Picture 6. The cargo insurance is also included in that calculation. This variable is determined according to distance and value of cargo. The Total Cost, represented by formula iv of Figure 11, is found with the result of the calculations of the Cost of Material (i) and the Cost of Conventional Freight (ii).

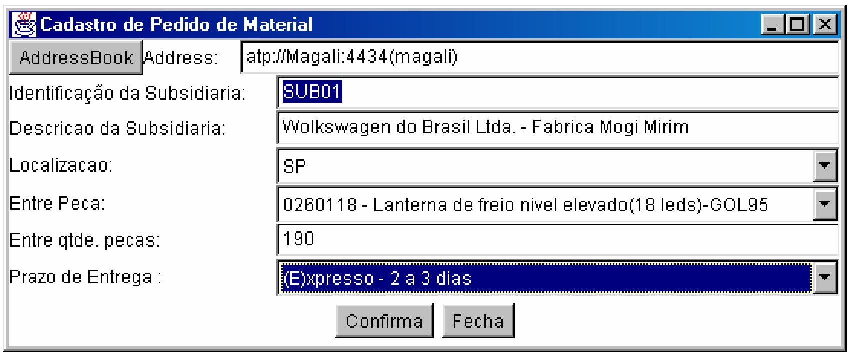
i) **MATERIAL COST** = Qtde_Req * CST_PC
ii) **CONVENTIONAL SHIPPING COST** =CFC + (Qtde_Req – 1) * CFUC + (insurance * MATERIAL COST)
iii) **EXPRESS SHIPPING COST** = CFE + (Qtde_Req – 1) * CFUE + (insurance * MATERIAL COST)
iv) **TOTAL COST** = MATERIAL COST + SHIPPING COST

Figure 11 - . The calculation of freight

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:** This Agent 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, as shown in Figure 12.



Cadastro de Pedido de Material

Address:	atp://Magali:4434(magali)
Identificação da Subsidiaria:	SUB01
Descricao da Subsidiaria:	Wolkswagen do Brasil Ltda. - Fabrica Mogi Mirim
Localizacao:	SP
Entre Peca:	0260118 - Lanterna de freio nivel elevado(18 leds)-GOL95
Entre qtde. pecas:	190
Prazo de Entrega :	(E)xpresso - 2 a 3 dias

Figure 12 List of Request of Material form

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, as shown in Figure 8. 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- Conclusions

After studying and elaborating this paper, we can conclude that the oriented programming paradigm to distributed agents is sufficiently matured to be used in system construction, where the IAD principles may be applied.

Regarding to the agents environment used in this work, Aglets Workbench, showed itself quite effective allowing the agents to travel on the web until the indicated server and making the research and/or local communication.

This paper mainly focused the two mobile multiagents development: Bon Marché and Logistic Agents.

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

The intelligent module implementation and the possibility of existence of more than a kind of product to be investigated are tasks to be implemented in another phase of this project.

Although the system finds itself in an experimental phase, where new ideas have been constantly estimated in the conceptual and the algorithm levels, the results obtained in this initial phase can be considered acceptable, due to the information filtering executed at the server where the resources are located, meaning , a significant reduced network loading.

Concerning the Logistics Agents system, a logistic model was implemented to solve the issue of automotive parts distribution, where each subsidiary and supplier generate 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 web to be overloaded.

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 Kabah’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.

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.Foner, L.N. (1997) *Entertaining Agents: A Sociological Case Study*, Proceedings of Automonous Agents `97, tbp

4.Franklin, S and Graesser, A.(1996) *Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents,* Proceedings of the 3rd Int. Workshop on Agent Theories, Architectures, and Languages, Berlin, Springer-Verlag

5.Further information can be found at http://java.stanford.edu/java_agent/html/

6.General Magic Inc. http://www.genmagic.com

7.Gilbert, D. and Janca, P. *IBM White Paper : Intelligent Agents*, http://www.networking.ibm.com/iag/iagwp1.html

8.Petrie, C.J. (1996) Agent-Based Engineering, the Web, and Intelligence, IEEE Expert, December 1996