Mobile Agents System for Knowledge Bases Processing

Claudiu POPÎRLAN and Mădălin FLORESCU

Faculty of Mathematics and Computer Science,
Department of Computer Science,
University of Craiova, Romania
popirlan@inf.ucv.ro, madalin.florescu@yahoo.com

Abstract. The importance of knowledge bases processing in Artificial Intelligence, virtual reality and computational vision was a challenging subject and a long motivation for researchers. The main idea of this paper is that complex knowledge bases can be processed using a mobile agents system. In this paper, a mobile agents system for knowledge bases processing is proposed. A Java application for knowledge bases processing is developed using mobile agents and Aglets Software Development Kit ([12]). Using the system, sophisticated knowledge have been represented and processed without increasing the application size, and with a very good computational time.
Keywords: mobile agents, knowledge base, Java application, knowledge processing
Math. Subject Classification: 68T30; 68T40; 68T05

1 Introduction

The idea of mobile agents approach to knowledge bases modeling is that complex knowledge can be processed with a with a very good computational time. The concept of mobile agent is defined in [1], [10]. They are autonomous objects that can migrate from node to node on behalf of the user who have executed them and make use of the databases or computation resources from clients connected by the network. In order for a mobile agent to be able to migrate, there must be a virtual place, the so-called mobile agent system, that supports mobility.

Knowledge bases representations have been discussed in [2], and are defined the Knowledge Representation and Processing System. Knowledge Representation using interactive network can be found in [6]. The idea of a good knowledge representation and processing [2, 5] is to use different system and mathematical functions to model knowledge. Based on this approach, the *semantic schemas* was proposed in [7, 8].

In this paper it is proposed a mobile agents system, intended very general and flexible, based on Aglets Software Development ([12]) effectively improve the problem of knowledge bases [2], [5] and distributed calculus [5] modeling.

2 The system architecture

Using the standard Java support we constructed and manipulated Java classes and objects inside a mobile agents application to make knowledge representation within the virtual world. The architecture of the system that includes the related components is described below and can be visualized in Fig. 1.

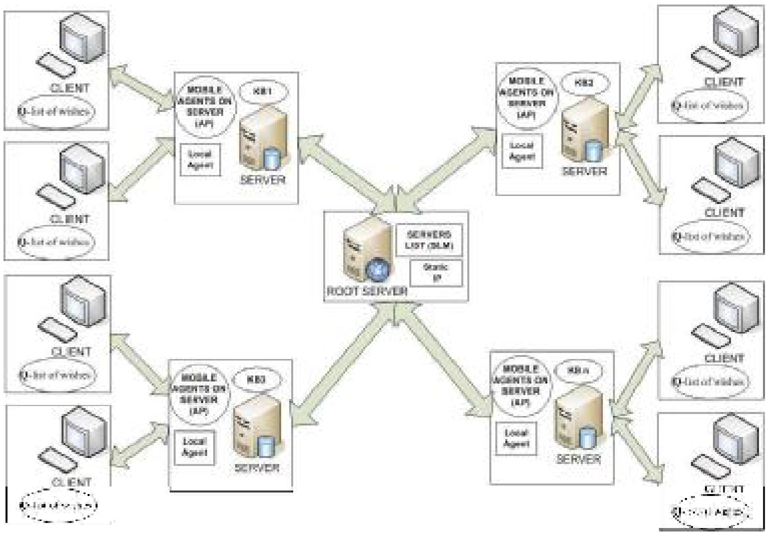

Fig. 1. The Mobile Agents System Structure

The role of the view-related components(defined terms) are as follows:

Definition 1 (AP). *The Agents Package (AP) represents all the agents that contribute to the knowledge base (KB) processing.*

Definition 2 (Server). *The Server is the computer that implements an agents package (AP) and a knowledge base (KB), and accept (can host / change information with) agents arrived from the outside.*

Definition 3 (Client). *The Client is a computer that implements an agents package (AP), and uses the system in order to query the knowledge base (KB).*

Definition 4 (Root). *The Root is a server whose static IP address is known by all the servers in the system. The presence of a knowledge base (KB) on this server is optional.*

The system consists of:

– n (n>0) number of Roots,
– any number of Servers and Clients,
– the agents package(AP),
– knowledge bases(KB1, KB2, .., KBn)

The system will be designed very general and flexible so as not to memorize specific information, but references to them.

2.1 The system functionality

The system must be able to process the knowledge base distributed on the servers that compose it. In this respect, there are Query Mobile Agents (QMA) which visit, one after another, all or a part of these servers to whom they ask for certain information. When an agent gathers all the knowledge specified by its user it returns home. The system must work independently of the server addresses. Thus, in a situation in which each server has a dynamic IP address assigned, the servers list maintained by the agent must also be dynamic. The solution for this list to reflect at any moment, as exactly as possible, the addresses of the servers is to constantly update it. When a server starts, it must announce its presence to be known within the system. At this point, the only address that it knows is the root one. Thus, the server sends to it a Presence Mobile Agent (PMA) with a double role:

– announces the root about its presence (as a server)
– gets from it a list of valid server addresses (optional, to avoid traffic load)

In this way, a new server can see all other existing servers and will also be known to all servers that will come later in the system. The problem is that existing servers can not see, under any circumstances, the emergence of the new servers. But this is necessary only when you want to create a QMA, which is why, each server will resend a PMA to the root, which may reconfirm its presence in the system and will return an updated list of servers, so that, once it is created and launched, a QMA will start with a list of valid server addresses. Another problem is that a QMA once learned in the process of visitation, has no way to know about the emergence of new servers or about those which were stopped. Thus, a QMA should be able to compare its own list of servers to that maintained by the server and update it if necessary. And here arises a problem because of the way in which a server updates its list: either when it creates a QMA, or when it is visited by a QMA. Assuming that none of the servers from the list of a QMA doesn't make any query nor is visited by other mobile agents, this (QMA) will not be able to update its list of servers and will not know about a possible appearance of a server that could help in its mission. In conclusion, there is one computer (root) who knows all the servers available in the system at any time. Any server that wants to create a QMA firstly obtains from the root a list of servers that are currently in the system and gives that list to the agent as a map. Any server that receives an agent must update its list according to that one from the agent (or vice versa). If a server has no activity during a certain period of time, thus having negative effects on how it succeeds in providing to the agents (QMA) a valid state of the system (list of available servers), it will resend a PMA, in other words, it will work on behalf of the system.

2.2 Looking information

The Query Mobile Agent (QMA) migrates through the network having a list of requests (or questions) set by the user on behalf of whom it is working, questions whose answer has to be found on one or more servers it visits. Once the agent gets all the solutions to its problems it can return home.

Let's imagine a representation of the wishes in the form of $K\ x$, where x is a positive number that uniquely identifies a desire (for agent) or a knowledge (for server). An agent member may have as its Wishes List (WL):

$$K\ 3\ ,\ K\ 18\ ,\ K\ 1\ ,\ K\ 10$$

A server might respond to the following knowledge, $K\ 18$, with::

$$K\ 20\ ,\ K\ 13\ ,\ K\ 7$$

When the agent queries the server, it will return the following new WL:

$$K\ 3\ ,\ K\ 1\ ,\ K\ 10\ ,\ K\ 20\ ,\ K\ 7\ ,\ K\ 13,$$

obtained by replacing knowledge K18 with new knowledge which will become new desires for agent. These new desires, along with those which does not yet have an answer will be the query for the next server to visit. As it is natural, understanding of a certain knowledge does not imply an understanding of another adjacent, related knowledge. An example of such knowledge is the fundamental geometrical notion of "point". From here arises the need to define knowledge in primary and complex. Therefore, an agent member may receive from the server, in response to its demands whether new extended knowledge that will be treated as new desires, or basic knowledge that will not require further answers. The mission of an agent is considered to be finished when it has only primary knowledge, and its list of desires is empty. An agent (QMA) has no way to know if the server it is going to visit may be useful or not (can provide answers to its requests), and moving on that server could mean just a waste of time and an unnecessary traffic load. There are two ways to decrease the browsing of an agent through the network:

- knowing those servers that have the greatest amount of information, or which gave the most answers to the agents which visited them

- knowing that an agent stores every knowledge that it finds, along with the corresponding address of the server that owns the knowledge, it could communicate those information to every server it visits.

Supposing that each server has an IPv4 address and the port number that it is listening on is always default (does not require memory for storing), it would mean that the list of servers is composed of many entrances to 4 bytes each. As you can see from the Table 1, for a very large number of servers is preferable for a mobile agent to carry only a few of them.

Table 1. Dependent number of servers and computational times

Number of Servers	Dimension of List	Transfer Speed	Computational Time
1.000	4 KB	32 kbps	1 s
1.000	4 KB	512 kbps	125 ms
1.000.000	4 MB	512 kbps	62.5 s
1.000.000	4 MB	1500 kbps	21.3 s

3 Mobile agents application for knowledge bases processing

For the knowledge bases representation, a mobile agents approach using object oriented programming (Java[11]) and Aglets Software Development Kit ([12]) is proposed.

The mobile agents application consists of three types of agents:

- Local Agent (LA)
- Query Mobile Agent (QMA)
- Presence Mobile Agent (PMA)

3.1 The system implementation

Regarding the implementation of the application components which are used by the user to represent and process knowledge bases, the next classes representing the basic structure of the Java application and described below have been used.

The classes structure of the Java application is shown in Fig. 2 and it is as follows:

- *KBM.agents*: This package contains classes that will create the agents
- *KBM.tools.SLM*: This package contains classes with role in the administration of the servers list for each of the three types of agents (LA, QMA, PMA)
- *KBM.tools.KLM*: This package contains classes with role in the knowledge list management for LA and QMA.

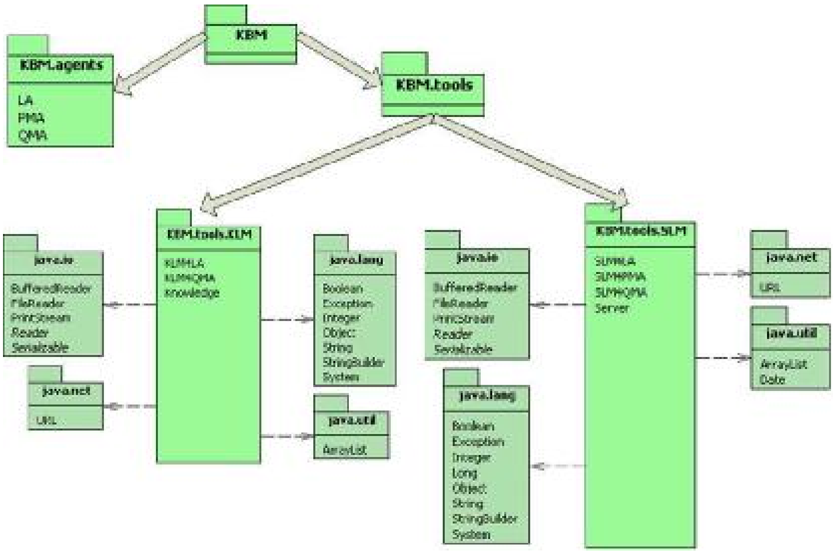

Fig. 2. The Class Diagram of the application

4 Results

The results from the mobile agents approach for knowledge bases processing using object oriented programming (Java) is presented. The Java application specifies the knowledge bases viewed as Java objects, allowing the mobile agents to process and to answer for diverse question. The important criteria of the application have been considered the easy implementation, functionality and a good running time. A very good computational time have been obtained per each system execution. Sophisticated knowledge bases have been represented and processed without increasing the file size.

For the following results, it is used the next knowledge base from the medical area(for diseases diagnose):

- *Level 1: Initial Knowledge: Symptoms:*
  $K1$ = wrist pain; $K2$ = swollen ligaments and tendons that form the carpal tunnel; $K3$ = wrist swelling; $K4$ = red eyes; $K5$ = light sensitivity; $K6$ = deformity of the wrist; $K7$ = discoloration of the skin;
-
- *Level 2: Extended Knowledge: Diseases:*
  $K10$ = sprain; $K11$ = carpal tunnel syndrome; $K12$ = arthrits; $K13$ = ganglion cyst; $K14$ = fracture; $K15$ = allergic conjunctivitis; $K16$ = dry eye; $K17$ = cyanosis; $K18$ = tendonitis;

–
– *Level 3: Primary Knowledge: Treatments*
$K20$ = heat; $K21$ = immobilization; $K22$ = ice; $K23$ = artificial tears; $K24$ = rest; $K25$ = compression; $K26$ = cortisone injection; $K27$ = anti-inflamatory medications; $K28$ = wrist splint; $K29$ = surgical procedure; $K30$ = cast; $K31$ = antibiotic drops; $K32$ = cloramphenicol ointment; $K33$ = tetracycline tablets;

Creating root server and local agents on each server. When local agents start there are sent to root PMA as shown in Fig. 3 We start four local agents: a root (LA1) and three servers (LA2, LA3, LA4).

Fig. 3. PMA3 is sent to the root

LA2 creates and sends QMA2 with a list of desires (K1, K3, K7). QMA2 visit each server, based on rating. Once arrived on the server LA4 it gets the following answers: K7 , K10, K17 then migrates on the LA3 server and obtains the following answers: K1, K3, K10, K11, K12, K13, K14, K18. When QMA2 exhausted the list of wishes it can return home (Fig. 4, Fig. 5).

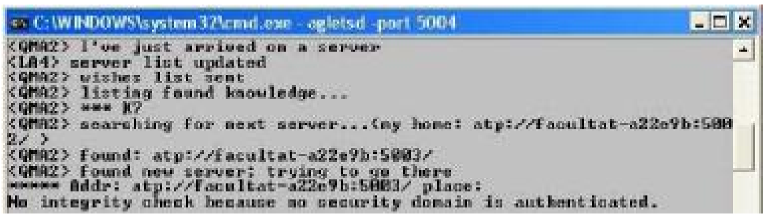

Fig. 4. QMA2 arrives on LA4

Once arrived home QMA2 will list the results in the form:

$$K\ x\text{: relevance: address}$$

Knowledge K, with the highest relevance, will be the answer to the original wishes of the agent(Fig 6).

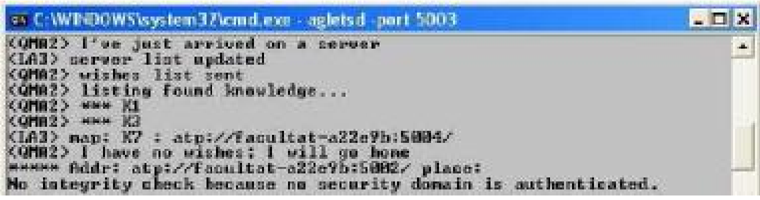

Fig. 5. QMA2 arrives on LA3

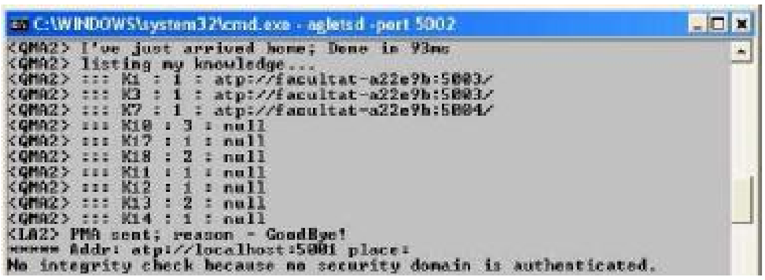

Fig. 6. QMA2 arrives home (LA2)

5 Conclusions

In this paper a mobile agents approach for knowledge bases processing using object oriented programming (Java) and Aglets Software Development Kit is proposed. A Java application which uses Aglets was developed. The suggested system indicates a method for knowledge base processing in a distributed environment. There are presented the stages of achieving the system and a running example of the system for a knowledge base in the medical area. As future direction of research, we shall try the implementation delivered the suggested model and the development of intelective internal model of agent, for the inclusion of mechanisms for processing and reasonings.

References

1. **P. Braun , W. Rossak**, em Mobile Agents: Concepts, Mobility Models, & the Tracy Toolkit, Elsevier Inc.(USA) and dpunkt.verlag(Germany), 2005
2. **N. Tăndăreanu**, Knowledge Bases, Universitaria Publishing House, (in Romanian), 2004
3. **Claudiu Popîrlan, Cristina Popîrlan**, Algorithms for Mobile Agents in Network using Tracy(Mobile Agent Toolkit), Research Notes in Artificial Intelligence and Digital Communications, Vol.106, 6-nd Romanian Conference on Artificial Intelligence and Digital Communications, Thessaloniki,Greece, August 2006, p.31-38

4. **Claudiu Popîrlan, Cristina Popîrlan**, Using Mobile Agents in User Interfaces Functionality, RoEduNet IEEE International Conference, Sibiu, Romania, 2006, p.337-340
5. **S. Russell and P. Norvig** Artificial Intelligence: A Modern Approach, Prentice Hall, 1995
6. **F. Leon, D. Galea, M. Zbancioc**, Knowledge Representation Through Interactive Networks, Proceedings of the European Conference on Intelligent Systems, Iasi, Romania, 2002
7. **N. Tandareanu**, Transfer of knowledge via semantic schemas, 9th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI 2005), Orlando, USA, July 10-13, Vol. IV, p.70-75, 2005.
8. **N. Tandareanu**, Semantic Schemas: The Least Upper Bound of Two Interpretations, 10th World Multiconference on Systemics, Cybernetics and Informatics (WMSCI 2006), Orlando, USA, July 16-19, Vol.III, p. 150-155, 2006.
9. **C.I. Popîrlan, & C. Popîrlan**, Mobile Agents communication for knowledge representation, 11-th World Multiconference on Systemics, Cybernetics and Informatics (WMSCI 2007), Orlando, USA, July 8-11, 1, 2007, 92-96.
10. **J. Baumann**, Mobile Agents: Control Algorithms, (Lecture Notes in Computer Science, Springer, 2000).
11. **B. Eckel**, Thinking in Java, Prentice Hall (4-th Edition), 2006.
12. ***, The Aglets Software Development Kit Webpage, (`http://www.trl.ibm.com/aglets/`).