Multi-Agent System for the Distribution and Location of Learning Objects

Lluvia E. Ponce[1], Karla Torres[1], Juan C. Ramírez[1], Fabiola López[2] , Darnes Vilariño[1]

[1] Facultad de Ciencias de la Computación, BUAP
Puebla, México
[2] Dirección de Modalidades Alternativas de Educación, BUAP
Puebla, México

pomell81@gmail.com, karlatorresmx@hotmail.com, jcramirezmx@gmail.com, darnes@cs.buap.mx, fabiola.lopez@siu.buap.mx

Abstract. The Learning Objects Paradigm (LO) is currently being used for the design of online courses with certain success. Nevertheless, the technologies that currently allow the sharing are not completely developed. In this paper, a system for the management of a distributed repository of Learning Objects is presented. This system considers the market as a Multi-Agent System –and it is named "Learning Objects Market (LOM)". In this market, Provider Agents enter to publish their LO; Consumer Agents request the LO that they need; and an Administrative Agent controls and monitors all of the transactions that occur inside the market. These actions guarantee the accomplishment of policies for login and permanence of the agents on the market. The interactions between all of the agents inside or outside of the LOM are presented, because depending on the type of service requested, the administrative agent throws a group of agents with particular tasks. An strategy and a negotiation protocol has been developed for both provider and consumer agents, on the process of buying and selling a LO. The results obtained with the first implementation of the LOM are also presented.

Keywords: Agents, Market, Learning Objects, Negotiation process.

1 Introduction

The quick advances on technology, the competitivity in the labour market and the need to be more prepared in a globalized world where the knowledge must be acquired with the least possible delay, are some causes for using new learning methodologies, in order to satisfy the demands for increasing the performance of a person in the process of teaching and learning.

Received 29/06/07
Accepted 19/10/07
Final version 24/10/07

The education leaves behind the traditional methods for teaching and starts with new learning systems. This creates new concepts, such as distance-learning, e-Learning, Virtual Learning Environments, etcetera. These new concepts generate new techniques for knowledge sharing, which shall be applied by professors and researchers around the world. They generate thousands of courses that are combined to produce content structures. These courses are created on web pages or e-Learning platforms, and are produced as monolithic. In other words, their content is not reusable once a course is divided; they lack of any interoperability because they are not multiplatform, and the production of these courses is slow and inefficient.

Most of the solutions to these problems is found on the Learning Objects Paradigm. A Learning Object (LO) is defined as "a digital or non-digital entity supported by technology which can be used, reused or referenced during the learning process" [1]. In this order, the courses are formed by LO and given their characteristics, the LO can be reused to generate new courses.

The use of the LO to generate courses introduce some problems such as: How to create them? How to store them? How to find them? How to use them? How to upgrade them?

The current paper is centered on the solution of two of these problems: the storage and the location. The development of an informatic system is proposed, which can publish, locate and control these objects. The system has been organized in a model of a Learning Objects Market (LOM).

The LOM has been implemented by following the Agent-Oriented Paradigm, and it is supported by the Services-Oriented Architecture (SOA) proposed by De Roure [2]. The basic components of a SOA are the following:

Provider Agents, which represent the interests of the LO developer users. They interact with the market to put on disposal several LO for Consumer Agents. The providers will keep certain rights and policies for access to the LO.

Consumer Agents, which represent the interest of course developer users. They dispose of tools to easily locate, access and negotiate use conditions for LO.

An Administrative Agent which regulates all the transactions made inside the market.

Before a LO is given from a Provider Agent to a Consumer Agent, they both must negotiate the conditions of use for the LO. Some of these conditions are price and permissions such as use, distribution and lending. This is done on a so-denominated Negotiation Stage.

The final agreements must be legalized in a contract that must be signed by both provider and consumer.

The paper is organized as follows: Section 2 describes the components of the LOM as a Multi-Agent System and the interactions of each agent inside the system. Section 3 contains the protocols and negotiation strategies that are used by the agents. Section 4 presents the communication language that allows to the agents to perform their tasks. Section 5 describes the JADE platform, which is used as the implementation tool for the agent behaviors. Finally, Section 6 gives the obtained results through the development of the system.

2 LOM Components as a Multi-Agent System

A general model of the market has been developed. This model shows the components of the SOA that have been implemented by following the Agent-Oriented Paradigm. To solve the previously presented problems, a Multi-Agent System has been proposed. It is composed by the following entities: Provider Agents (PA), Consumer Agents (CA) and an Administrative Agent (AA).

With the purpose of attending simultaneously the queries made by the different PAs and CAs, the AA tasks have been divided in a group of subtasks. Each of the subtasks is performed by independent agents -such as register agents, login handler agents, LO Localizer Agent and Contract Register Agent. These agents are conveniently created by the AA depending on the service requested by any entity of the market. It is important to remark that, based on the proposed model, there are several agents that 'reside' inside the market, and others that are external to the market. Both kinds of agents are listed next.

Resident Agents of the LOM:

Administrative Agent (AA). This agent creates the LOM, creates the agents involved on the MAS, manages the LOM, delivers LO to the CA and waits for queries sent by the Mini-Reporter Agent to create CA inside the LOM.

Interface Agent for the AA (IAAA). This agent waits for actions made by users, and access remotely to the Market Data Base to send response to the user queries.

Consumer Agent. This agent registers on the LOM, locates LO, negotiates about the use of determined LO and logs out of the LOM.

Register Agent (RA). This agent registers to both PAs and CAs to enter and exit the LOM.

Login Handler Agent (LHA). This agent publishes LO, retires LO, and allows to PAs and CAs to drop out of the market.

LO Localizer Agent (LOLA). Locates to the LO Provider Agents.

Contract Register Agent (CRA). Delivers a contract for the use of a LO.

External Agents to the LOM:

Interface Agent for the CA (IACA). This agent waits for actions of the consumers, shows agreed contracts and policies for login and permanence inside the market.

Mini-Reporter Agent (MRA). This agent communicates with the market, by representing a consumer.

Provider Agent. This agent registers in the LOM, publishes LO, negotiates contracts for use of a LO and delivers LO.

Interface Agent for the PA (IAPA). This agent waits for actions of the providers, shows agreed contracts and policies for login and permanence inside the LOM, and keeps the key to access to the LO.

The proponed Multi-Agent System is shown on the Fig. 1.

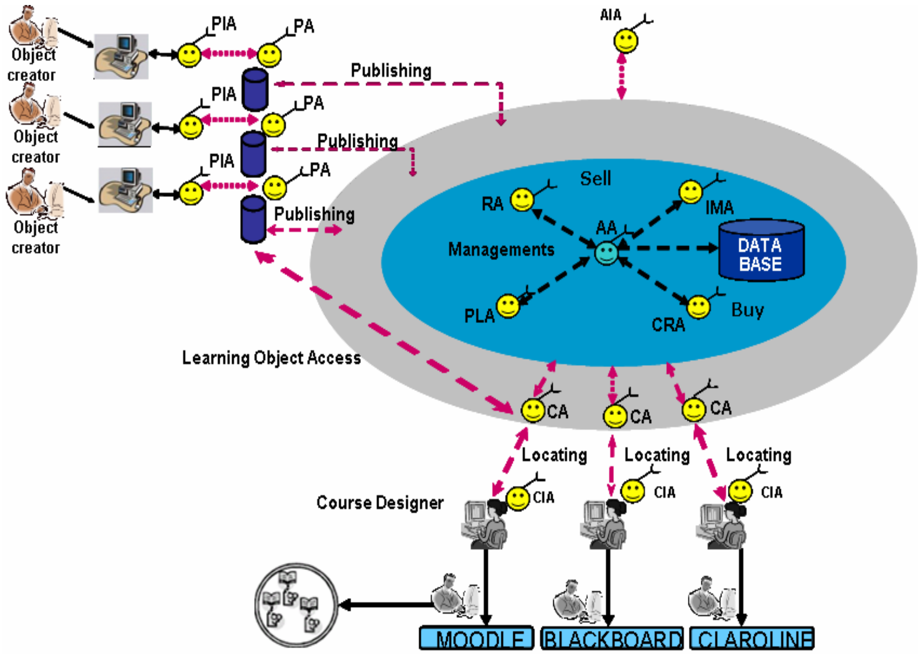

Fig. 1. General Model of the Learning Objects Market

2.1 Interactions of the Multi-Agent Systems

In order to obtain a good performance of the LOM an interaction process between the agents should exist. Due to the tasks performed by each agent are not isolated, in other words, the execution of a task implies the finishing of a previous task. The interaction process is described next.

The user that creates a LO is represented by a PA that performs the tasks described on the previous section. The developer puts on disposal of the customers their objects through these agents. Both the PA and the user that creates LO interact with the IAPA to maintain communication. The PA and the IAPA do not form the market; generally they are located on a remote node. Any action or request made by the user that creates LO must be delivered to the PA through the IAPA, and the PA will then communicate with the market and eventually, the PA will communicate with the IAPA to deliver to the user the obtained results of these actions and queries.

The consumer user of LO is represented by a CA that is in charge of obtaining the LO requested by the user to create his own courses. Both the CA and the consumer user interact between them via a IACA. The CA resides inside the LOM while the IACA is located on the computer of the consumer user, so the communication between them is made remotely. Any action or query performed by the consumer user is delivered to the CA via the IACA, and the CA will communicate directly to the market. Eventually, the CA will communicate with the IACA to deliver to the user the results obtained of these actions or queries.

The objects that are given or published by the PAs are stored on a global repository. The control and management of the LO is in charge of the AA.

The AA is always in execution state. Due this agent is in charge of receiving requests from the PAs and CAs, it verifies the kind of request and creates the corresponding agent to the performed task, it may be a RA, RHA, LOLA or CRA, to attend the query.

Regarding to the social structure of the market, it may be possible that the PAs and CAs have different interests about how to obtain or offer a LO. So before a LO may be delivered, a process of negotiation about the use of the LO must be initialized. The process will conclude with the sign of a contract that includes the terms and conditions of use of the LO. The duties, benefits and penalties in case of breach of contract are also established in this contract. The PAs and CAs will be provided with capacities of dynamic negotiation. The core of these agents is described next, in other words, the communication language, strategies and protocols that are used on the Negotiation Stage.

3 Negotiation

On Garima's research [3], a model of the negotiation agent is presented. This agent is able to operate in different environments and to adapt dynamically to them, due to the architectural division of the agent in modules. In the current paper, the model proposed by Garinma is used as a base for the design of an agent able to accomplish dynamic negotiation.

Conceptual model of the agent. An agent is composed by an static core and interchangeable components. Each agent is composed of three basic modules (see Fig. 2).

Communication Module. This is responsible of the communication between agents. This is a static module that supports the ACL FIPA language.

Protocol Module. It contains general rules for the negotiation. When an agent starts the negotiation, based on the negotiation table, it decides which negotiation protocol can be used, and dynamically it loads the correct module.

Strategy Module. It contains policies, a set of goals, actions and action rules. The chosen strategy depends on the success rate and the selected communication protocol. If there is not available information for these conditions, a default strategy model is uploaded to the agent.

As it is presented on the Fig. 2, an agent is capable to dynamically choose any appropriated combination for Protocol and Negotiation Strategy for a particular negotiation case. The result of applying these combinations may be stored for making future decisions. Then, in the moment of choosing a combination Protocol-Strategy that is considered to bring a successful negotiation, the agent may use the acquired experience, or apply a different combination that has never been applied or with a low success ratio. This is done to avoid a predictable performance that may be applied against it by other agents.

The following Protocols and Strategies are proposed to be included in the agent programming:

Protocol Module: Monotonic Awarding Protocol (MAP) and Rubinstein's Alternate Offers Protocol (RAOP).

MAP. The MAP works following these steps:

1. The negotiation goes in rounds.
2. On the first round, the agents simultaneously propose a deal of the group of negotiations.
3. The agreement is reached once the deal of an agent is as good as its proposal.
4. If the agreement is not reached, then another round of simultaneous proposals proceeds.
5. Any agent must propose a deal that is worst than a previously proposed deal.
6. If any agent makes an award in any round, then the negotiation ends with a conflictive agreement.

Using the MAP, the negotiation is guaranteed to end -with or without agreements- after a finite number of rounds.

RAOP. There are N agents, $Agents = \{A_1, ...A_n\}$. The agents need to reach an agreement about a given situation. It is assumed that the agent may take actions on the negotiation only in certain moments of possible periods of agreement $T = \{0,1,2,...\}$. These periods are determined and known in advance by the agents. On every period $t \in T$ of the negotiation, if it has not previously finished, the agent makes its offer in the time t. In this moment, there may occur an agreement –respect to the situation specified on the negotiation. Each of the agents may accept the offer (Yes), reject it (No) or leave the negotiation (Out). If the offer is accepted by all of the agents, the negotiation ends and the offers are implemented. If at least one of the agents decides to get out of the negotiation, it ends in conflict. If all of the agents stay in the negotiation, but at least one rejects the offer, the negotiation continues on the period t+1. The rejecting agent makes a counter-offer, the rejected agent responses, and the negotiation continues.

Strategy Module. In the negotiation context, the LOM has two agents: the PA that wants to sell its LO to the best possible price, and the CA that wants to acquire a LO to the best possible price.

The intervals that the price of the LO may take have been defined. The negotiable parameters considered by the PA are the following:

1. Desired price. It is the amount that the LO creator wants to obtain.
2. Minimal acceptable price. It is the lowest price that the LO creator will sell its LO.

Because of these parameters, the PA has to respect the following parameters for the price variation, defined on the range:

$$[p\_minp, p\_desp] \quad (1)$$

The negotiable parameters considered by the CA are the following:

1. Desired price. It is the price that the consumer is disposed to pay for the LO.
2. Maximum acceptable price. It is the highest price that the agent is disposed to pay for the LO.

Due to these parameters, the CA has to respect the following parameters for the price variations, defined on the range:

$$[p\_desc, p\_maxc] \tag{2}$$

Three strategies for negotiation have been designed and implemented. Each proposal made by the PA or the CA -by following the previously described protocols- is found on the defined ranges, respectively for the kind of agent.
The proposed strategies to be included in the concerning module are the following:

Strategy of Constant Increases-Decreases.
Strategy of Increases-Decreases by a Parabolic-Function.
Strategy of Priority Negotiations by Increases-Decreases.

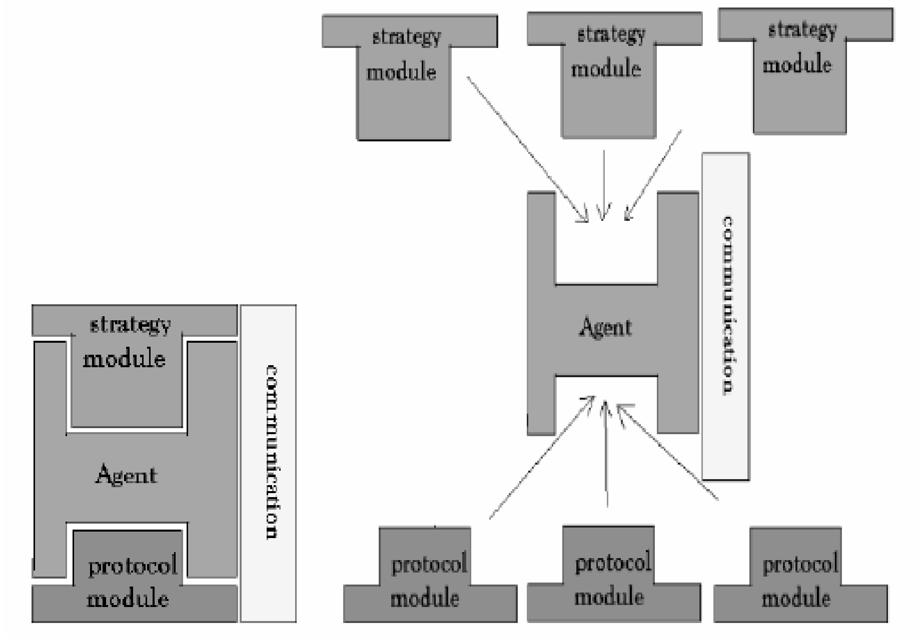

Fig. 2. General Model of the Learning Objects Market

4 Communication between Agents

In order to perform the negotiation and the tasks of the agents involved in the LOM, a comprehensive communication language between them is needed. An agent may reject a query of another agent, so they must be capable to speak between them, do decide which actions to perform or which data to obtain.

ACL has been chosen as the communication language between the agents.

An ACL message is a communicative act that an agent delivers so the receiver performs certain action.

The classical syntax is a LISP type, with several parameters such as the following.

1. About the communication. They contain the sender and receiver and intermediate entities.
2. About the content. Such as the language, ontology, etcetera.
3. About the intention, such as what message is replying, which is the following protocol, etcetera.

This can be expressed in XML or any other descriptive language.

ACL has a group of communicative acts. On the following Table, the strategies performed for negotiation and communication between the agents and the LOM are presented.

Table 1. Communicative Actions with ACL.

Communicative Act	Meaning
Cfp	Request a proposal.
Refuse	Reject a query.
Not-understood	The message is not understand.
Propose	Make a query.
Reject-proposal	Reject a proposal.
Accept-proposal	Accept a proposal.
Inform	Give information to the receiver.
Failure	Explains why the action failed.
Request	Asks to perform an action.
Agree	Accepts query.
Cancel	Cancels query.
Query-if	Asks something to the receiver.
Subscribe	Subscribe to a service of a receiver.

5 Implementation of the Multi-Agent System

The MAS has been developed in the JADE platform ((Java Agent DEvelopment Framework). This platform is implemented in Java. Jade simplifies the implementation of Multi-Agent Systems through a middleware that follows the FIPA specifications. It also contains a set of graphic tools which help on the debugging and deployment phases. The platform of the agent can be distributed into different hosts, they do not share the same operative system and the configuration can be controlled by a remote graphic interface.

5.1 Agents Behavior in JADE

A behavior represents a task that an agent can achieve. The behaviors are classified in three groups:

1. One-shot behavior: Behaviors that complete immediately and whose action() method is executed only once.
2. Cyclic behavior: Behaviors that never complete and whose action() method executes the same operations each time it is called.
3. Generic: Behaviors that embed a status and execute different operations depending on that status. They are completed when a given condition is met.

It is necessary that the implementation of one of these behaviors inherits from the jade.core.behaviours.Behaviour class, to get an agent to achieve an implemented task for a behavior object. It is enough to add the behavior to the agent through the addBehaviour() method of the Agent class. Behaviors can be added at any time, when an agent starts –setup() method, or in another behaviors [4]. Any class that inherits of Behaviour must implement the action() method where are described all the types of behaviors that an agent use to execute each one of its tasks.

Each agent of the LOM has its own tasks. They are implemented based on some of the previous behaviors that were mentioned above. As an example, the implemented behaviors of the administrative agent and the types of behaviors that the agent uses to achieve each one of its tasks, are described next.

1. AnalizaMensajesBehaviour: These behavior responses queries of PA or CA. It is a cyclic behavior because a query can arrive there at any time.
2. AnalizaRequestBehaviour: These behavior responses queries of the internal agents of the market. The behavior is similar to the latter.
3. CrearConsumidoresBehaviour: This behavior creates CAs in the market. They received the name of MRA. The behavior is one-shot type because for each consumer user there is a CA.

In the similar way the other behaviors of agents have been implemented.

6 Conclusions and Recommendations

The assigned tasks of each agent that is in the MOA have been fully implemented. To reach our objective, the communication protocols and all of the messages that are used in the interaction process have been defined, with the main purpose to achieve the goals of the agents.

According to the Negotiation phase, the MAP and the RAOP have been implemented. The latter one, that considers the strategies of constant increments and decrements makes these movements with a quadratic function.

Currently the first version of the LOM has been implemented in JADE. This middleware has shown an excellent performance. The Multi-Agent System can be consulted in the following web page: www.dgmae.buap.mx/invdes/mercadoOA.

The developed agents senses the environment, they take the information and then send and receive messages according to the external data. This task implies the implementation of new behaviors. The authentication process achieved in the LOM is the key access validation; an agent receives a key once it is registered in the market, and however it is not enough to maintain the security in the process of MOA. Now we are working in authentication of PA and CA, another task to implement is the process of sending and receiving messages in a secure way to avoid the losses of information.

Acknowledgements. Research financed by the Trust SEP-UNAM, Project 159/2006.

References

1. IEEE 1484.12.1, Learning Technology Standards Committee, Learning Object Metadata Standard.
2. D. De Roure, N. Jennings, N.R. Shadbolt., The Semantic Grid: A future e-Science Infraestructure.hnical report UKeS-2002-02.
3. P. Marcin., A. Abraham1, Agents Capable of Dynamic Negotiations. Computer Science Department Oklahoma State University Tulsa, OK, 74106, USA.
4. G. Caire, JADEProgramming-Tutorial-for-beginners, JADE Documentation, 2003, p. 8.