IE 2010
Congreso Iberoamericano de Informática Educativa
Jaime Sánchez, Editor
Santiago, Chile, 2010

A MULTIAGENT SYSTEM FOR PERSONALIZED RECOMMENDATION OF LEARNING OBJECTS

Ana Casali, Valeria Gerling, Claudia Deco and Cristina Bender
Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario
Argentina
acasali@fceia.unr.edu.ar, valeria.gerling@gmail.com, {deco, bender}@fceia.unr.edu.ar

ABSTRACT
In this paper a multiagent Educational Recommender System is presented. The purpose of this system is to select the best learning objects from a federation of repositories according to characteristics and preferences of each user. This system has a multiagent architecture and one of its main agents, the Personalized Search Agent (PS-Agent), is modeled as a graded BDI agent. The graded BDI agent model allows us to specify an agent architecture able to deal with graded mental attitudes. We focus on the implementation aspects of the recommender system and especially on the PS-Agent development. Also a case study, which shows promising results in learning objects ranking, is presented.

Categories and Subject Descriptors
I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems.

General Terms
Design, Experimentation.

Keywords
Recommender System, Information Retrieval, User Needs, Education.

1. INTRODUCTION
In the last years, Artificial Intelligence community has carried out a great deal of work on recommender systems [11]. This kind of systems can help people to find out what they want, especially on the Internet. These systems take personal preferences into account, and infer and intelligently aggregate opinions and relationships from heterogeneous sources and data. Furthermore, we want such systems to be scalable, open, privacy-protecting and we want to get the recommendations with the least possible work on users' behalf. Thus, agent technology makes it possible to specify distributed, complex and autonomous recommender systems.

Among recommender systems we particularly concentrate on the educational domain, because there is a great amount of diverse resources that can contribute to the teaching-learning process. Moreover this is an interesting domain, where user's preferences and restrictions need to be considered. Because of the variety of possible needs, recommender systems can be modeled at different levels of complexity and knowledge-based approaches appear to be very suitable [1]. In this work we focus on learning objects recommendation, where a learning object (LO) is "any digital resource that can be reused to support learning" [6]. LOs can be used by a student who wants to learn a subject, or may be used by a teacher who wants to prepare materials for his/her class. LOs are described with metadata usually in the standard LOM (http://www.ieee.org). Users can retrieve LOs through searches in web repositories. Examples of such repositories are: FLOR (http://www.laclo.org), Ariadne (http://www.ariadne-eu.org), and OER Commons (http://www.oercommons.org).

Given a topic query, a user has as result the same list of LOs. Generally, he/she checks only the top results, but in many cases these top results are not suitable if the search is performed considering only topic keywords. This is because users have different characteristics and preferences, which should also be considered at search time. Recommender systems arise to solve this kind of problem because they can select the material that is most appropriate to user's needs and preferences. Our approach is to achieve the customization of search results taking into account LO metadata, with semantic descriptions, and a user profile, including characteristics and preferences.

Respect to recommender systems in education, Zhu et al. [14] propose a multi-agent architecture of personalized recommendation system, which can provide personalized services for learners and instructors. Six software agents coordinate work hierarchy with each other to offer functions including personalized recommendation. Wang et al. [13] propose an adaptive personalized recommendation model in order to recommend SCORM-compliant learning objects. They use a hybrid method that recommends LOs, using two algorithms: preference-based and correlation-based. Lu [10] presents a personalized learning material recommendation framework and develops two related technologies: a multi-attribute evaluation method to justify a student's need, and a fuzzy matching method. García Salcines et al. [8] present a collaborative recommender system that uses distributed data mining for the continuous improvement of e-learning courses. It allows teachers with similar profiles, to share their research results as a result of applying data mining locally on their own courses. In this work, we follow a

Casali, Ana., Gerling, Valeria., Deco, Claudia., Bender Cristina. (2010). A Multiagent System for Personalized Recommendation of Learning Objects. En J. Sánchez (Ed.): Congreso Iberoamericano de Informática Educativa, Volumen 1, pp 503-510, Santiago de Chile.

multiagent content-based approach, and we take into account the user profile, the LO metadata and domain knowledge to obtain a recommendation.

Respects to agent technology, several architectures have been proposed to provide agents with a formal support. Among them, a well-known intentional formal approach is the BDI architecture proposed by Rao and Georgeff [12]. This model is based on the explicit representation of the agent's *beliefs (B), desires (D)*, and *intentions (I)*. The agent´s beliefs represent all the information the agent has about the environment, the agent´s desires are the states of the world the agent wants to reach (the agent´s desires may be ideal and sometimes unachievable). Some of these states are the goals the agent is committed to achieve; these are the agent's intentions. Indeed, this architecture has evolved over time and it has been applied, to some extent, in several of the most significant multiagent applications developed up to now.

A more flexible BDI architecture to design and develop agents potentially capable of having a better performance in uncertain and dynamic environments has been proposed in [3, 4]. In this work the authors have proposed a general model for graded BDI (g-BDI) agents specifying an architecture able to deal with the environment uncertainty and with graded mental attitudes. In this agent model, *belief degrees* represent to what extent the agent believes a formula is true. *Degrees of positive or negative desires* enable the agent to set different levels of preference or rejection respectively. *Intention degrees* give also a preference measure but, in this case, modeling the cost/benefit trade off of reaching an agent's goal. Consequently, agents having different kinds of behavior can be modeled on the basis of the representation and interaction of these three attitudes. This agent architecture has solid logic formalization and has been used previously to design and implement a tourism recommendation agent [5].

In this work we present the development of a recommender system of learning objects where one of its main agents, the *Personalized Search Agent (PS-Agent)*, is modeled as a graded BDI agent. The system goal is to recommend learning objects from a federation of repositories, according to user's characteristics and preferences. A preliminary and Spanish version of this work has been proposed in [2]. Now, we focus on the system prototype implementation and in addition, a case study is presented. This paper is organized as follows: in Section 2 the g-BDI model of agent is given. Then, in Section 3 the architecture for the Recommender System of Learning Objects is proposed. In Section 4 we described the design of the PS-Agent. In Section 5 we present the implementation of the Recommender System prototype. Next, in Section 6 we analyze a case study and finally, we present some conclusions and future lines of work.

2. g-BDI AGENT MODEL

The architecture proposed in [3, 4] is specified using multi-context systems (MCS). This approach is suitable to represent complex logical system, allowing to represent in different contexts local logical aspects and then, combining them by inter-context rules. The MCS specification contains three basic components: units or contexts, logics, and bridge rules, which channel the propagation of consequences among theories. Thus, an agent is defined as a group of interconnected units: $\langle\{C_i\}_{i\in I},\Delta_{br}\rangle$, where each context $C_i \in \{C_i\}_{i\in I}$ is the tuple $C_i=\langle L_i,A_i,\Delta_i\rangle$ where L_i, A_i and Δ_i are the language, axioms, and inference rules respectively. When a theory $T_i \in L_i$ is associated with each unit, the specification of a particular MCS is complete. The deduction mechanism of these systems is based on two kinds of inference rules, internal rules Δ_i, and bridge rules Δ_{br}, which allow embedding formulae into a context whenever the conditions of the bridge rule are satisfied.

In the g-BDI agent model, we have at least mental contexts to represent beliefs (BC), desires (DC) and intentions (IC). This model also considers two functional contexts: for Planning (PC) and Communication (CC). In summary, a g-BDI agent model is defined as:

$$A_g = (\{BC, DC, IC, PC, CC\}, \Delta_{br}).$$

The overall behaviour of the system will depend on the logic representation of each intentional notion in the different contexts and the bridge rules. In order to represent and reason about graded notions of beliefs, desires and intentions, the agent model uses a modal many-valued approach. For instance, let us consider a Belief context where belief degrees are to be modelled as probabilities. Then, for each classical formula φ, we consider a modal formula $B\varphi$ which is interpreted as "φ is probable". This modal formula $B\varphi$ is then a fuzzy formula which may be more or less true, depending on the probability of φ. In particular, we can take as truth-value of $B\varphi$ precisely the probability of φ. Moreover, using a many-valued logic, it can express the governing axioms of probability theory as logical axioms involving modal formulae. Then, the many-valued logic machinery can be used to reason about the modal formulae $B\varphi$, which faithfully respect the uncertainty model chosen to represent the degrees of belief. It has been set up an adequate axiomatization for the belief context logic combining axioms for the different formulae. The same many-valued logic approach is used to represent and reason under graded attitudes in the other mental contexts. The formalization of the logics for the different contexts is described in [3].

3. RECOMMENDER SYSTEM ARCHITECTURE

We present the architecture of a recommender system of learning objects where the system's goal is to recommend Learning Objects (LOs) from a federation of repositories, according to user's characteristics and preferences. We found very suitable to design this system using a multi-agent architecture because it can manage heterogeneous and distributed information, such as LOs Repositories, and have a high degree of modularity and autonomy. Furthermore, these systems are highly scalable and open. This system includes several types of agents according to their different roles. The architecture proposed is shown in Figure 1 and some details of the agents' functionalities are described next.

The *Interface Agent (I-Agent)* interacts with the user through a graphical interface. It captures data submitted by the user, and displays search results. This agent provides user preferences and restrictions to the UP-Agent. Also it provides the search topic to

SR-Agent. Then, it receives from PS-Agent the recommendation (ranking of LOs) that it will be given to the user and finally, receives the user feedback.

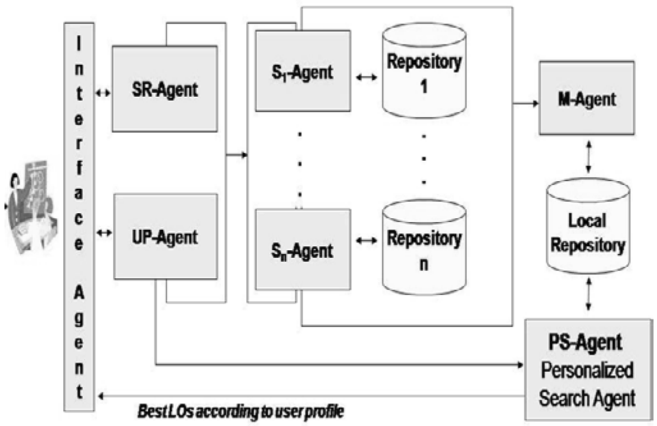

Figure 1. Multiagent System Architecture

The *Semantic Refiner Agent (SR-Agent)* produces the search strategy, using as input the set of terms that describe the topic of interest, disambiguating them and expanding each term by incorporating synonyms and semantically related terms [7]. The output of this agent is a search strategy that will be provided to each search agent.

The *User Profile Agent (UP-Agent)* receives user data, user preferences and restrictions from the Interface Agent, in order to build his/her profile. This agent provides to the Search Agents (S_i-Agent) some restrictions (called primary restrictions) that can collaborate to filter LOs, for example, if the materials have or have not cost. In addition, this agent provides to the Personalized Search Agent preferences and remaining restrictions of the user, so that the objects are properly sorted.

Each *Search Agent (S_i-Agent)* searches, in a LO repository, objects that match user search strategy and some of the primary restrictions. For this, it adapts the search strategy to the syntax of the repository to which it accesses and then it establishes a communication with the repository through a communication protocol. Each result is parsed, transformed into a common format (e.g. XML), and sent to the Mediator agent which stores this result in a local repository.

The *Mediator Agent (M-Agent)* integrates these results and solves conflicts in order to have consistent data. Finally, it stores the information in the local repository.

The *Personalized Search Agent (PS-Agent)* takes this information and produces an ordered list of recommended objects and it is described in some detail in the following subsection.

4. DESIGN OF THE PERSONALIZED SEARCH AGENT

The Personalized Search Agent (PS-Agent) has been specified using the g-BDI model presented in Section 2. This agent model is suitable for the PS-Agent, since it allows to represent in an explicit way graded positive and negative preferences (i.e. as the agent's desires) and uses them to conduct the agent to the best intention (i.e. the LO more suitable for the educational goal). The g-BDI agent model has contexts to represent the agent beliefs (BC), the agent desires (DC) and the agent intentions (IC), and we have chosen an appropriate reasoning model to deal with graded attitudes in the different contexts. We describe the principal characteristics of these contexts and give an example of a bridge rule in the PS-Agent.

Belief Context (BC): This context represents the information the PS-Agent has about education environment, including characteristics of the LOs. These characteristics are described by metadata according to LOM standard and encoded in XML. From the LOM metadata set, we have selected the most relevant to the recommender agent such as Language, Learning Resource Type, Interactivity Level, Intended End User Role, Context, Difficulty, Typical Learning Time and Cost. These metadata will be considered by the agent to infer the degree of preference satisfaction namely, to compute the belief degree b_{ik} with that an object O_i is expected to satisfy a particular user preference p_k. This is represented by the formula $B(O_i, p_k, b_{ik})$. For example, the belief degree in a LO satisfaction of the user's preference that the resource would be "practical", considering that its *type of resource* is specified as [exercise, lecture], is computed as 0.6 in the BC and is represented by B(O_i, style=practical, 0.6).

A set of rules have been established for each kind of preferences (e.g. interaction, role, learning style, language, etc.) and an adequate distance is used for each case (for a complete description of these rules the reader is referred to [9]). Next, we present an example of rules for some kinds of preferences:

- *Interaction*

  (R1) IF InteractivityLevel(O_i)= 'low'

  THEN B(O_i, interaction=low, 1)

  (R2) IF InteractivityLevel(O_i)= 'high'

  THEN B(O_i, interaction=low, 0.2)

- *Role*

  (R3) IF IntendedEndUserRole(O_i)=[learner,. . .]

  THEN B(O_i, role=learner, 1)

  (R4) IF IntendedEndUserRole(O_i)=[a,b,c, teacher,. . .]

  and a,b,c $\in$ {learner, author, manager}

  THEN B(O_i, role=teacher, 0.4)

- *Learning style*

  (R5) IF LearningResourceType(O_i)=

  [exercise,narrative text,slide]

  THEN B(O_i, style=mixed, 1)

  (R6) IF LearningResourceType(O_i)=[exercise,lecture]

  THEN B(O_i, style=practical, 0.6)

Desire Context (DC): The global desire of the PS-Agent is to find the LO that best fits the user profile, taking into account the selected subject, restrictions and preferences. In this context we represent the user preferences (e.g. language or academic context) and the restrictions (e.g., maximum duration) that he/she wants for the LOs. Preferences and restrictions may be graded (with values, called priorities, in [0, 1]) expressing respectively different levels of preference and rejection. An extract of an example of user profile is shown in Table 1.

Preference	Priority	Description
Mother language="Spanish"		Mother language
Language="Spanish"	1.0	Document languages that the users wants.
Language="English"	0.8	
Language="French"	0.6	
Interaction="low"	0.8	Degree of interaction the user wants
Restriction	**Priority**	**Description**
Max-duration="60"	0.8	Maximum time that the user expects it takes to work with the object.

Table 1. Extract from a user profile

This table shows that the user strongly prefers that the LO language should be Spanish (degree equal 1), also he is satisfied with a resource in English (degree equal 0.8) and may be in French with a lower degree of preference (0.6). Furthermore, the user desires, with degree 0.8, that the interactivity with the LO should be low. Regarding restrictions, the user indicates with priority 0.8, that he/she does not want objects which duration exceeding 60 minutes. Formally, this is represented by:

$$D^+ (language = Spanish, 1)$$

$$D^+ (language = English, 0.8)$$

$$D^+ (language = French, 0.6)$$

$$D^+ (interaction = low, 0.8)$$

$$D^- (Max\text{-}duration = 60, 0.8)$$

Intention Context (IC): Intentions are the educational goals that the user will try to reach through the best object (or objects) selected. It is expected that the user's learning process will be improved by the LOs that fit best his/her profile.

The intention to reach the goal (of satisfying a set of preferences $P=\{p_k \mid k = 1 \ldots n\}$) by a particular O_i, has a degree r_i and is represented by the formula $I(P, O_i, r_i)$. To compute r_i the agent consider each preference p_k degree d_k and the expected satisfaction of this preference through an educational resource O_i, represented by b_{ik}. These variables are combined through an appropriate bridge rule (see rule (1)) that using belief and desire formulas, determines the degree of intention r_i of each O_i to satisfy all the preferences.

$$\frac{DC: (D^+ p_1, d_1), \ldots, (D^+ p_n, d_n); BC: B(O_i, p_1, b_{i1}), \ldots, B(O_i, p_n, b_{in})}{IC: I(P, O_i, r_i)} \quad (1)$$

This degree of intention is then used to rank the LOs in the agent final recommendation.

The function f to compute the degree of intention r_i associated with each O_i, $f(d_1, \ldots, d_n, b_{i1}, \ldots, b_{in})$, may be defined in different ways. For example, in our prototype implementation we have defined it as the average of the expected satisfactions of all the user´s preferences.

$$r_i = \frac{\sum_{j=1}^{n} d_j \times b_{ij}}{n} \quad (2)$$

Other factors may be included in the bridge rule (1) as, for example, the cost of the resource (if there is any) or the reliance on the source of the learning object (e.g. institution, author, etc.). Also, different functions can be used to compute the intention degree, modeling in this way different behaviors of the PS-Agent.

In the following section we present a simple example to illustrate the different features and interactions of the agents in the multiagent system.

4.1 Example

Suppose that José is an engineering student who is looking for documents with information about Matrix as he is studying a first course of Algebra. On the other hand, he has great knowledge of English, and understands some French. José wishes that the learning style should be practical. He wants that the time that takes the development of the material will not exceed 60 minutes and also, that the selected object will has no cost.

When José performs the search, he provides as input the word "*matrix*". The Semantic Refiner Agent interacts with the user and constructs the associated search strategy. Then, the User Profile Agent builds the profile of José through an input form. From this interaction, his profile contains among other data, the information shown in Table 2.

Suppose now, that the Search Agents based on the resulting search strategy and the primary restrictions (Cost = "no"), retrieved from the repositories a set of four learning objects (O_1, O_2, O_3, O_4) with their metadata. After the Mediator Agent solves potential conflicts and integrate data, these LOs are stored into a local repository. Table 3 shows the most relevant metadata (LOM standard) of these retrieved objects O_i, $i \in \{1 \ldots 4\}$.

Preference	Priority
Mother language="Spanish"	
Language="Spanish"	1.0
Language="English"	1.0
Language="French"	0.6
Role="learner"'	1.0
Interaction="low"	0.7
Learning style="practical"'	1.0
Academic context="university"	
Knowledge level ="initial"	1.0

Restriction	
Max-duration ="60"	0.7
Cost="no"	

Table 2. Extract user profile of José

LO	O_1	O_2	O_3	O_4
Lang.	English	Spanish	French	English
Learning Resource Type	lecture	[exercise, lecture]	slide	exercise
Interactivity Level	low	low	very low	high
Intended End User Role	learner	learner	learner	teacher
Context	higher education	higher education	school	higher education
Difficulty	medium	easy	medium	difficult
Typical Learning Time (min)	40	50	20	50
Cost	no	no	no	no

Table 3. Relevant metadata of the objects recovered

All these objects satisfy the user's needs and the restriction of having no cost. Now the PS-Agent starting from the remaining restrictions and preferences will select those that are most appropriate for José. First the agent applies the rule of maximum duration restriction. As each O_i, $i \in \{1 \ldots 4\}$ satisfies this condition and also, the language of each one is selected by the user (at different level of priority), then the four LOs are considered for the ranking.

The PS-Agent using the rules in the Context Belief (defined in Section 4) calculates the expected satisfaction of each user's preferences (e.g., interaction = low, language = Spanish) through the different characteristics of each O_i. From these rules, the PS-Agent compute the belief degree b_{ij} with that an object O_i is expected to satisfy a particular user preference p_j, $B(O_i, p_j, b_{ij})$.

Then PS-Agent calculates the intentions to reach their goal by applying the bridge rule (1) and the function defined by the equation (2).

Computing these values for the four objects, the following degrees of intention are obtained:

$r_1 = 0.72$ $\quad r_2 = 0.86$ $\quad r_3 = 0.6$ $\quad r_4 = 0.608$

Finally, the PS-Agent sorts the objects by decreasing the value of the degree obtained: O_2, O_1, O_4, O_3. Then the system recommends to José the course O_2 as the most suitable and also gives an ordered list of the other alternatives.

5. PROTOTYPE IMPLEMENTATION

A preliminary prototype of the Recommender System composed by the Personalized Search Agent (PS-Agent), the Interface Agent (I-Agent) and one Search Agent (S-Agent) has been implemented. For the development of this prototype it was used the languages SWI-Prolog and Ruby on Rails. The PS-Agent is implemented using SWI-Prolog because it is an open source logical programming language. Moreover, it is multithread allowing representing in a suitable way the agent logical architecture and deductions. Ruby is used to implement the I-Agent and the S-Agent, since it supports the ability to consume Web Services with SOAP, which is necessary to communicate with the repositories using the SQI protocol.

Through a graphic interface the user introduces the current preferences and restrictions in his/her profile. Figure 2 shows the prototype user interface (it is in Spanish language as it is for regional purpose).

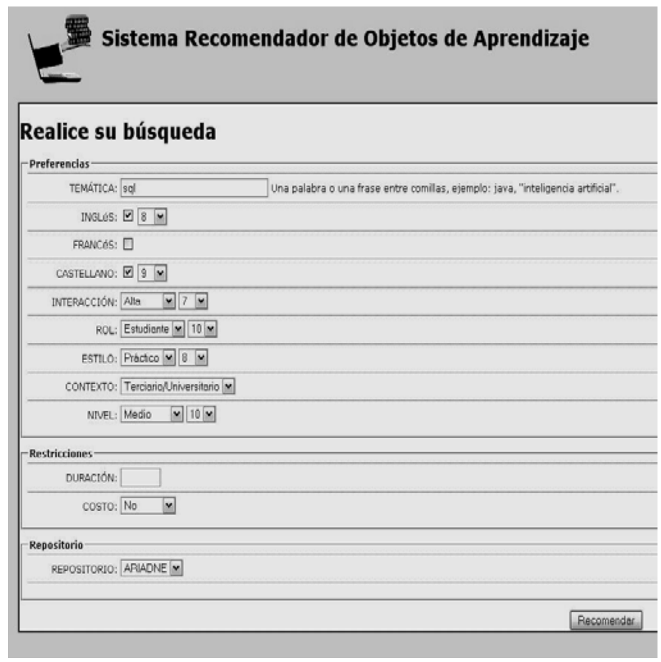

Figure 2. User Interface; subject, preferences and restrictions of the LO

The Interface Agent (I-Agent) acquires and saves in a data file the information of the user's search. This data file stores also the user search history, keeping the chosen subject and his/her preferences (languages, role, interaction, etc.), for statistical post processing. Then, the I-Agent communicates the subject of interest to the Search Agent (S-Agent) and the information related to preferences and restrictions to the Personalized Search Agent (PS-Agent).

The first search of LO (i.e. a subject search) is made by the S-Agent. The S-Agent sends a query according to the selected subject to a repository (or federation of repositories) and waits for a response. In this first prototype we use the repositories Ariadne and FLOR. The S-Agent can communicate with these repositories through the SQI protocol and the query language VSQL. As response, the S-Agent receives an XML file containing the metadata, according to the LOM standard, of each of the LOs that satisfy the selected subject. Furthermore, the S-Agent creates an XML file for each of these objects, and stores them in the local repository. Also, it creates a file with the number of objects and the name of each XML file describing each object. Then, this file is communicated and used by the PS-Agent to generate the ordered list that the Interface Agent presents to the user.

After receiving from the S-Agent the resulting LOs retrieved from the repositories, the PS-Agent selects the more suitable ones and provides them to the I-Agent. For achieving this, the PS-Agent generates a file with the ranked results and displays the top 10. For each of the selected LO, it is shown its identification (ID) and the following characteristics: title, description and location. Figure 3 illustrates a result recommendation together with the details of one of the selected LO.

Figure 3. Result of the recommendation and detail of the tenth object

After receiving the recommendation the user can express its opinion through the interface selecting one of the following three options:

- *Correct*: when the ranked list of objects satisfies the user.
- *Different order*: when the user is in accordance with the selected objects but the order does not match his preferences. Then, the user can indicate which he/she considers the best top three.
- *Incorrect*: when the user is not satisfied with the list of recommended objects. Then, the interface enables him to introduce a textual comment.

In this prototype there exists a user with an administrator profile. This administrator can view the search history of all users and their personal data and, in order to evaluate the system performance, he/she can access to different statistics obtained from the stored data. Example of the use of these data will be seen in the case study presented in Section 6.

6. CASE STUDY

To test the recommender system, Ariadne repository was used, since it has a high percentage of loaded LO educational metadata. This allows the recommender to build the ordered list of LOs that satisfy the user profile. For this case study, users were students and professors of the Computer Science career of our University.

Tables 4 and 5 show different user profiles according to the information captured by the Interface Agent. User characteristics of interest are his/her Mother Language and Subject to search. Also, the interface captures user priorities for: *different languages* (French, English and Spanish), *Role, Interaction, Style, Context, Max-duration*, and captures if the user wants LO with or without *Cost*. The value for Role may be teacher or learner. Interaction refers to the level of interaction the user prefers (i.e. low, medium, high) expressing his preference about the LO. Style value may be theoretical or practical. The Context (university, school, etc.) has also a Level (advanced, medium, initial). All these different kinds of preferences may have a *Priority*, an integer number in [0, 10], expressing different level of preferences.

Profile	**User1**	**User2**	**User3**
Mother language	Spanish	Spanish	Spanish
Subject	java	java	java
Language Spanish Pr	10	10	10
Language English Pr	9	10	10
Language French Pr	0	9	9
(Role,Pr)	(l,10)	(l,10)	(l,10)
(Interaction,Pr)	(medium,5)	(ignore,-)	(ignore,-)
(Style,Pr)	(theo,9)	(theo,9)	(p,9)
(Context,Level,Pr)	(u,i,9)	(u,i,10)	(u,m,10)
(Max-duration,Pr)	(0,-)	(0,-)	(0,-)
Cost	no	no	no

Table 4. Profiles of User1, User2 and User3

Profile	**User4**	**User5**	**User6**
Mother language	Spanish	Spanish	Spanish
Subject	java	java	java
Language Spanish Pr	10	10	10
Language English Pr	10	10	10
Language French Pr	9	9	9
(Role,Pr)	(t,8)	(t,8)	(t,7)
(Interaction,Pr)	(ignore,-)	(high,7)	(ignore,-)
(Style,Pr)	(theo,10)	(theo,10)	(p,10)
(Context,Level,Pr)	(u,i,9)	(u,a,9)	(u,i,7)

(Max-duration,Pr)	(0,-)	(0,-)	(90,9)
Cost	no	no	no

Table 5. Profiles of User4, User5 and User6

For space reasons, the notation we use is as follows: Priority (Pr), university (u), initial (i), medium (m), advanced (a), learner (l), teacher (t), theoretical (theo), practical (p).

We decided to search LOs about Java, because this topic is known for these users and also Ariadne has free accessible material on this topic in languages known by these users. The search filtered 27 of 263 objects, leaving only those LOs about java whose language was English, French or Spanish. The Recommender System offers to each user an ordered list of objects according to his/her preferences. For example, for User2 the system recommends the following ordered list of LOs:

$$\{Id_{12}, Id_{11}, Id_{13}, Id_7, Id_6, Id_8, Id_4, Id_{15}, Id_{26}, Id_2\}$$

Table 6 shows the most relevant metadata of some of these retrieved objects, sorted by object identification. The first column contains the LOs metadata of interest: Language, Learning Resource Type (the kind of LO), Interactivity Level (the degree of interactivity characterizing this LO), Intended End User Role (principal role for which this LO was designed), Context (is the principal environment within the use of this LO is intended to take place), Difficulty (how hard it is to work with or through this LO for the typical intended target audience), and Typical Learning Time (in minutes). To reduce space, the notation we use is as follows: French (fr), English (en), unknown (unkn), medium (med), and education (educ).

After receiving the ranked list, the user can give his/her opinion about the recommendation results. User2 was satisfied with the objects but the order does not match exactly his preference and suggests *Different Order*, proposing that the first three objects should be:

$$\{Id7, Id6, Id8\}.$$

To evaluate proximity between PS-Agent recommended ranking and the user's own ranking, we use Manhattan distance, which is suitable to capture distances between positions. This distance was calculated between positions of the first three items ordered by the user with the corresponding positions in the recommended list, taking into account the degree of intention of each object in the list. Table 7 shows distances calculated on the five successful cases.

LO	Id_6	Id_7	Id_8	Id_{11}	Id_{12}	Id_{13}
Language	fr	fr	fr	en	en	en
Learning Resource Type	slide	slide	slide	slide	slide	narra-tive text
Interactivity Level	unkn.	unkn.	unkn.	med.	high	unkn.
Intended End User Role	learner	learner	learner	learner	learner	learner
Context	higher educ.	higher educ.	higher educ.	higher educ.	higher educ.	higher educ.
Difficulty	unkn.	unkn.	unkn.	med.	med.	unkn.
Typical Learning Time (min)	90	135	90	60	90	3

Table 6. Relevant metadata of some of the objects recovered

User	Distance	Worst Case
User2	6.0	66.0
User3	14.0	55.0
User4	9.0	66.0
User5	25.0	67.0
User6	14.0	72.0
Average	**13.6**	**65.2**

Table 7. Distances computed from successful cases

User1 is excluded because he/she evaluated the response of the recommender as *Incorrect* and therefore, there is neither an order suggested nor the distance may be computed.

PS-Agent performance depends mainly on the completeness of LO metadata in the repository and on the quality of this information. For example, we can see that some of the LO metadata in Table 6 has the value unknown (for the metadata Interaction level and Difficulty). Taking this problem into account and considering that we have obtained a distances average of 13.6, with an average of 65.2 in the worst cases, we can say that in most cases the list of recommended objects is not so distant from the user´s one. Thus, we can claim that the preliminary performance of this recommender system gave us promising results. However, results may not satisfy a user since, although a document satisfies his/her preferences set in the interface, this specification may not fit what really the user want, because there are subjective factors that are difficult to capture and model.

Analyzing this case study we found some difficulties associated with metadata information. In some cases, an incorrect classification of metadata was found. For example, a document had metadata *Language=English*, but it had only the title in English and the document body was in French. Also, we found metadata that classified correctly a document from the point of view of the file type, but they do not adequately describe the document contents. For example, a document had metadata *LearningResourceType=narrative text*, but it was actually a Java program to solve the traveling salesman problem; so it should be better classified as an *exercise or example*. This classification of the resource as *narrative text* makes that the application of rules has the highest degree of belief that the object satisfies the theoretical style, which is not true.

7. CONCLUSIONS AND FUTURE WORK

In this paper we have presented the architecture and implementation of an educational recommender system. This architecture is based on a multiagent system which allows work with flexible, scalable, heterogeneous and distributed information from LO repositories. In particular, we designed the Personalized Search Agent (PS-Agent) as a graded BDI agent since it is suitable to deal with user's graded preferences. A prototype was implemented using Ruby on Rails for Interface Agent and Search Agents, and SWI-Prolog for the PS-Agent. The case study has shown promising results in LOs ranking. In this first stage, users evaluated whether objects were useful and if the recommended order was correct, according to his/her opinion. One problem for the experimentation was the lack of information on many of educational metadata of the learning objects in the repositories. Another difficulty presented was that many objects could not be accessed. Currently, we are working in the integration of the recommender system as part of an assistant to help teachers to assemble LOs according to an instructional design, considering their students characteristics and preferences. In another line of research, we are working on the automatic extraction of metadata.

8. ACKNOWLEDGMENTS

The authors acknowledge partial support by the Project JARDIN LACCIR-RFP2008 (Latin American and Caribbean Collaborative ICT Research and Microsoft) and Project 219308 SECTEL, Gobierno de Santa Fe.

REFERENCES

[1] Burke, R. 2000. Knowledge-based recommender systems. *Encyclopedia of Library and Information Systems*, 69.

[2] Casali, A., Gerling, V., Deco, C. and Bender, C. 2009. An Intelligent System to Assist the Personalized Search of Learning Objects. In *Revista CyT. Universidad de Palermo*, Vol. 9. Buenos Aires, diciembre 2009. 113-127.

[3] Casali, A., Godo, L. and Sierra, C. 2005. Graded BDI Models For Agent Architectures. J. Leite and P. Torroni, Eds. *CLIMA V, LNAI* 3487, 126-143.

[4] Casali, A., Godo, L. and Sierra, C. 2009. g-BDI: A Graded Intensional Agent Model for Practical Reasoning, *Modeling Decisions for Artificial Intelligence, 6th International Conference*, MDAI 2009. Lecture Notes in Artificial Intelligence, Vicenç Torra, Yasuo Narukawa, Masahiro Inuiguchi, Eds. Vol. 5861, Awaji Island, Japan, Springer, (November 30, 2009) 5-20.

[5] Casali, A., Von Furth, A., Godo, L. and Sierra, C. 2008. A Tourism Recommender Agent: From theory to practice. In *Revista Iberoamericana de Inteligencia Artificial*, AEPIA, Vol 12:40 (2008), 23-38.

[6] Chiappe, A., Segovia, Y. and Rincon, Y. 2007. Toward an instructional design model based on learning objects. In *Educational Technology Research and Development*. Boston: Springer, 671-681.

[7] Deco, C., Bender, C., Saer, J., Chiari, M. and Motz, R. 2005. Semantic refinement for web information retrieval. In *Proceedings of the 3rd Latin American Web Congress*. IEEE Press. 106-110.

[8] García Salcines, E., Romero, C., Ventura, S. and de Castro Lozano, C. 2008. Sistema recomendador colaborativo usando minería de datos distribuida para la mejora continua de cursos e-learning. *IEEE-RITA* 3(1): 19-30.

[9] Gerling, V. 2009. *Un Sistema Inteligente para Asistir la Búsqueda Personalizada de Objetos de Aprendizaje*. Degree Thesis on Computer Science. National University of Rosario. DOI=www.fceia.unr.edu.ar/lcc/t523/tesina.php?campo1=21.

[10] Lu, J. 2004. A Personalized e-Learning Material Recommender System. In *Proceedings of the 2nd International Conference on Information Technology for Application (ICITA 2004)*. China, 374-379.

[11] Montaner, M., López, B. and de la Rosa, L. 2003. A Taxonomy of Recommender Agents on the Internet. In *Artificial Intelligence Review* 19: 285-330, Kluwer.

[12] Rao, A. and Georgeff, M. 1995. BDI agents: From theory to practice. In *Proceedings of the 1st International Conference on Multi-Agents Systems*, 312-319.

[13] Wang, T.I., Tsai, K.H., Lee, M.C. and Chiu, T.K. 2007. Personalized Learning Objects Recommendation based on the Semantic-Aware Discovery and the Learner Preference Pattern. *Educational Technology & Society*, 10 (3), 84-105.

[14] Zhu, F., Ip, H. and Cao, J. 2008. PeRES: A Personalized Recommendation Education System Based on Multi-agents & SCORM. In *Advances in Web Based Learning (ICWL 2007)*, LNCS 4823/2008, 31-42.