International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:3, 2014

A Recommender Agent to Support Virtual Learning Activities

P. Valdiviezo, G. Riofrio, R. Reategui

***Abstract*—**This article describes the implementation of an intelligent agent that provides recommendations for educational resources in a virtual learning environment (VLE). It aims to support pending (undeveloped) student learning activities. It begins by analyzing the proposed VLE data model entities in the recommender process. The pending student activities are then identified, which constitutes the input information for the agent. By using the attribute-based recommender technique, the information can be processed and resource recommendations can be obtained. These serve as support for pending activity development in the course. To integrate this technique, we used an ontology. This served as support for the semantic annotation of attributes and recommended files recovery.

***Keywords*—**Learning activities, educational resource, recommender agent, recommendation technique, ontology.

I. Introduction

In general, many virtual courses today offer copious resources and tools that serve as support in online learning processes. The vast amount of resources registered in some course offers indicates that we should examine other didactic strategies that offer more personalized attention by means of technological tools. These learning support tools help reduce the excessive burden of teaching responsibilities. It is important to know; therefore, what resources are available and which resources are the most adequate means for carrying out a specific activity. In order to solve this problem, we require a recommender agent that supports the capacity of developing tasks, such as suggesting new topics and recommending activities and services according to user priorities [1]. One study shows that there are two ways of achieving this: first to predict if a product is going to be accepted by users and second to recommend products based on the likes and preferences of users [2]. The application of this concept in the educational area and particularly in open educational resources implies that the learning environment ought to be able to recommend resources and activities based on student learning preferences and needs. Recommender systems in [3]'s work discusses personalized and real-time assistance by harnessing resources according to student requirements in an educational environment, thereby developing learning activities.

An important activity when recommending resources in virtual courses is to choose the most adequate resources to solve the proposed activities in a specific course. In this case, it is necessary to determine the relationship with the task being carried out and to identify those resources that contribute to the development of such an activity.

Therefore, when designing recommender systems, it is important to establish the goals of such a system. Once the recommender system objective is established, it is important to think about the most adequate techniques for its development. This is achieved by reviewing current theories of recommender systems. There are many different techniques as can be seen in [4]-[6].

It is important to select the most adequate techniques when considering the usage of the proposed agent in the virtual learning environment (VLE). This paper begins with a theoretical revision of how recommender systems are evaluated in the educational context and their functionality. Different recommender techniques are also reviewed. Then, data from the entity-relational model of the VLE are analyzed together with the course activities performed by students. Finally, we present the proposed agent architecture and the results of the implementation in the VLE.

II. Recommender Systems in an Educational Context

Recommender systems or agents today constitute a useful alternative for providing personalized support to system users. They are also being used for providing feedback to the user [7].

In [8], there is a description of user experiences with recommender systems. It covers the essential elements of the definition of a recommender system and information about how and when to implement it. This information can be determined after analyzing student feedback. Another model is proposed in [9], which describes learning objects in a personalized recommender system for students using e-learning systems.

In e-learning scenarios, recommending strategies can be proposed in order to suggest activities to the students. For example, incorporating learning activities, consulting educational material or using any other resources that enable students to enhance and improve the learning process.

Generally speaking, recommendation scenes are established by setting rankings to those items that are visited by specific users. The items with the highest ranking are then selected for making the recommendation. Therefore, a recommender system can be defined as a group of users represented by *C*, and *S* – where the latter is represented by a group of items that are to be recommended (in our case these items would be the educational resources), and a utility function, which is represented by *u*, which determines the usefulness of the item to the user: $u: C \times S \rightarrow R$, *R* is an ordered group of integers or

P. Valdiviezo, G. Riofrio, and R. Reategui work at the Universidad Tecnica Particular de Loja, DCCE, 1101608, Ecuador (phone: 593-072570275; e-mail: pmvaldiviezo@utpl.edu.ec, geriofrio@utpl.edu.ec, rmreategui@utpl.edu.ec).

non-negative real numbers that are in a certain rank [5]. That is why each user belongs to *C*, and an item belongs to S, thus maximizing the usefulness to the user. This usefulness can be defined by scoring (ratings) and the user preference measure for the item. In our study, the ratings can be replaced by the semantic relationship between the resources and the activities, which are carried out by the students.

Nevertheless, in an educational environment, the complexity of the item is due to the vast amount of knowledge and learning activities related to the courses in the virtual environment, as well as the different preferences for specific resources or services, etc.

Thus, in [8] there is a model for generating recommendations in an educational environment. This research work selects those elements that comprise the recommender agent, namely:

Category: This is related to the scope of the recommender system. Some categories can be established according to the context and the objectives of the recommender system. In this case, the category is determined by the objective of the recommender agent, thus providing recommendations to support learning activities.

Technique: Technique is the essential part of the recommender system since it is used to maximize the usefulness of each item for the user. Although there are several techniques and classifications, the most adequate ones are selected for this case. Among the different recommender techniques available, the one based on attributes was selected. For the sub-type of this technique, we used the one based on knowledge.

Origin: The origin determines the source of the recommender system; it is linked with the category and depends on the context where the recommender system is used.

Applicable Conditions: These are related to the environment characteristics and determine whether it is necessary to propose a recommendation or not. The conditions for this would be determined by the interaction between the student and the program and the availability of educational resources in the university course. The recommendations are provided when the student enters the course and reviews the pending activities.

A. Recommender Techniques

The recommender systems help to filter and present information, which depends on both the technique that is being used and the students' preferences. In this way, a personalized service is achieved. Furthermore, it makes it easier to explore the system.

There are various classifications of recommender techniques that focus on the previous methods [5]. The requirements, techniques and the initial model of a personal recommender system for lifelong learning students are described in [10]. In this work, the authors propose a combination of personalized recommender techniques for learning activities in the context of e-learning.

Based on what has been reviewed in previous work, these recommender techniques can be classified as follows:

Content Based: The systems using these types of algorithms are designed to recommend items that consider the individual characteristics of elements [11].

Collaborative Filtering: This enables recommending items to be utilized by users with similar profiles [12], likes, and preferences. These systems make it possible to identify elements based on similar users' opinions for making recommendations. However, this technique is not the most adequate for the proposed agent as it requires users to rate several items, that is, to generate a recommendation and ensure that the program adheres to the recommendations with similar likes.

Hybrid Systems: These combine both techniques: collaborative filtering and content based [13].

There is another classification based on the algorithm used for doing the prediction, which is:

Model Based: The model based systems use information according to the user and establish a model to generate recommendations.

Memory Based: This technique uses all the available information to generate the recommendation.

For this work, it was necessary to select the best technique for our context. Taking as a reference research papers [4], [6], [10], it could be determined that the memory-based techniques are the most adequate for the educational environment. An analysis of the techniques belonging to this classification was performed whereby the most suitable would become the attributes based on knowledge. Consequently, they could be applied to the agent because they facilitated the inclusion of non-related characteristics and helped to map the user needs of the item (resources). In [4], the capacities of these techniques are developed by determining several sub-types and establishing the knowledge-based technique such as sub-types of the attribute based technique– both of which are convenient for the recommender agent.

III. Design Solutions

The proposed recommender agent aims to provide support to students involved in the Virtual Learning Environment at the Universidad Técnica Particular de Loja (UTPL), that is via the recommendation of educational resources that can be used to develop learning activities that have yet to be carried out by the students (pending activities). Within this context, the first phase is the analysis of the virtual learning environment and the identification of pending activities.

For the recommender process it is necessary to have a semantic component and information recovery techniques, which aid semantic annotations, the extraction of information about pending activities and the inference of educational resources that are recommended to students. The agent is implemented in the virtual learning environment via web services, which facilitate communication with the components of the proposed architecture.

IV. Review of the Virtual Learning Environment at UTPL

The Virtual Learning Environment at the UTPL provides support to the students in the distance education program. This VLE is based on Moodle, which has been adapted to the institution's needs.

In the VLE, each course has an announcement area with different topics. This is also where the student activities can be added. There are several options for uploading student activities in VLE, including questionnaires, forums, resources, and exercises.

An Open Educational Resources (OERs) section was incorporated in the VLE.

When using the VLE, specific tasks and forums are selected by the teacher. These are the ones that the UTPL teacher should incorporate into his/her classes:

Tasks: these are proposed by the teacher. Students can't see the other students' tasks unless they are in a group; different task formats can be uploaded.

Forums: these facilitate interaction and discussion between the teacher and students, and also between other students. There are different types of forums, but the most common is the open forum where the teacher suggests a topic and then students interact online and post their answers.

It is important to identify which teacher proposed activities and which activities have not yet been developed by the student. When the teacher does this, he/she can decide which tasks should be carried out and also recommend some useful resources.

V. Pending Activity Detection in the VLE

In order to identify pending student activities two different courses were used as a sample. These courses corresponded to the April - August 2013 academic period. The students' work was then monitored. Next, we determined the activities using the available tools and services of the VLE. To analyze the students' online activities, we examined information from the main 'entities', thus linking students, courses and activities. This enabled us to determine the number of times the student participated in a learning activity like a forum or a task. The registered entities refer to forums (mdl_forum_discussions) or tasks (mdl_assignment_submissions) that were subject to modifications.

In this sense, the intervening entities in this interaction are users, courses, tasks and forums. We observed the relationship between the user table containing the student information and those registering the students' participation in forums and course activities.

After that, we determined whether a student participated in tasks or forums. These were registered in the tables: mdl_forum_discussions and mdl_assignment_submissions. Consequently, they are identified as student pending activities.

To remind the students about the pending activities for the course, it was necessary to create a section in the VLE to display these online activities for students.

Additional support was provided by means of learning resource recommendations for the above-mentioned activities, so as to contribute to the development of course activities and to improve the learning process by providing appropriate resources.

That is why a recommendation technique based on the attributes of these activities is required. Besides, a semantic focus helps establish the relationship between the educational resources and the pending activities.

A. Ontologies Used as Recommendation Support

For pending tasks, semantic annotations are required to describe the recommended files. For this, a KIM1[1] tool ontology called KIMO was used, where new entities and files were saved according to the annotations.

KIM has several APIs which are used by the recommender agent. APIs facilitate semantic annotations and recommendations. Moreover, KIM offers a complete set of knowledge management tools.

After considering KIM's characteristics, the next step was to integrate this technology with the recommender agent so as to include the semantic element. This was necessary for recovering files and for describing the pending tasks in the VLE courses

VI. Recommender Agent

According to [14], an agent can be described as an entity capable of performing a specific task using or processing well-structured information. The agent structure is then presented with the intervening components to generate recommendations.

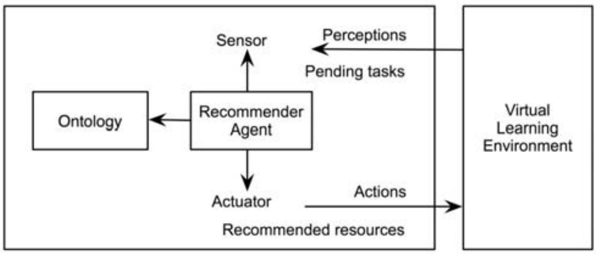

Fig. 1 Recommender agent schema

The characteristics of this scheme are described below:

Environment: This refers to the environment where the agent works. In this case, it is the VLE and where the recommendations' interface is utilized.

Perceptions: These are the perceptions of the agent comprising the stored information log files. They contain information about the pending activities carried out by the students and the educational resources that are going to be recommended.

Actions: In this case, the recommendations are the resources to be used for developing the activity. The actors are the elements proposed by the agent containing the results, which are then used to process the information. Here, the interface

[1] http://www.ontotext.com/kim

represents the location where the agent displays the educational resource recommendations.

Moreover, the agent uses a Kim platform for semantic processing and metadata information transformation.

VII. Proposed Recommender System Architecture

Based on the recommender system models and techniques for educational environments reviewed in the state of art, we proposed a scheme to provide a global view of the functional agent using the following architecture:

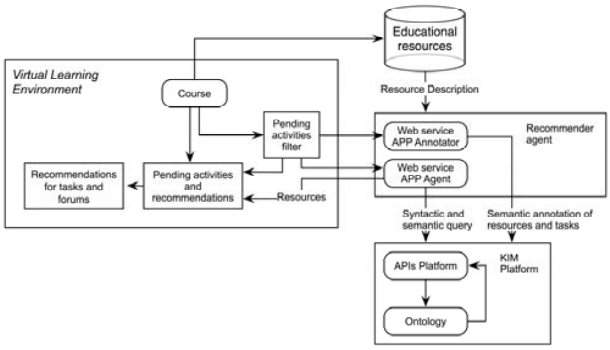

Fig. 2 Overview of the architecture of the UTPL recommender system

The above architecture shows how the agent interacts with VLE and the KIM platform to develop the semantic components. In this example, the VLE provides information about the students' pending activities and details related to the educational resources.

During this stage, the agent makes the semantic annotations and recommendations using the KIM platform and two web services.

For the annotation component, the agent describes both the pending activities and the educational resources to be saved in the KIM repository. It saves both the entities and the documents. The web service that performs this task is called the 'annotator'.

For the recommendation component, the web service recommender agent intervenes in this process. After making the semantic annotations, it does a semantic search in the OERs entities and recovers them with the aim of offering an online resource recommendation for the various student activities.

The elements comprising this architecture are:

A. Pending Activities Module

Here, all the pending student activities are detected. That is why a module has been created to display these activities for the student. This module is added to the interface that displays the resource recommendations for each pending activity.

B. API -KIM Platform

In order to accomplish the semantic annotation function and ensure the information recovery, it is necessary for the KIM applications to be accessed by means of the two previously-mentioned web services.

The Kim platform APIs to be accessed are the following:

- Semantic Annotation API service: This is integrated with the recommender system with the aim of invoking semantic annotation processes. Two main methods can be distinguished.
- Document Repository API service: This saves every file that is recovered with methods like loadDocument or getDocument.
- Semantic Repository API service: Its function is to record the semantic information like RDF. The service allows users to search and to modify the repository.
- Entity API service: It allows users to integrate the whole semantic information of a given entity with only one call.
- Query API service: By means of this service, we can submit complex queries both to the semantic repository and to the files repository. It combines a query function of the Document Repository API and the Semantic Repository API, where semantic and syntactic queries can be performed.

C. Annotator and Agent_recommender Web Service

A data table is shown below with the web service methods:

TABLE I
Web Service Annotator

Input Data	Method	Output
Server (server information).	Connect	Returns the rmi variable, which is used to access the Kim services by means of a series of methods.
Data (resource information).	ER Processing	This method sends the resources to the annotation method.
rmi, item, content and tasks and resources. information mode.	Annotate	This method uses the KIM functions to annotate and save files
Task	Recommend	This method sends the tasks to the KIM "anota_task" method.

This service acts like an agent. Its function is to make semantic annotations of every pending activities and every open educational resource. In addition, it saves both entities and documents in the KIM repository.

The following table describes another web service used by the recommender agent:

TABLE II
Web Service Recommender Agent

Input Data	Method	Output
Server (server information)	Connect	Used to access the APIs of the platform: it returns a KIM variable, which is used to access the APIs of the platform.
Activity (VLE activity information)	Recommend	This method uses a semantic search of documents related with the annotated entities of the tasks and recovers them in a data arrangement. To transform the objects of one type to another in php, we have to use json. A result is not obtained; it is sent as a syntactical query.
Annotations	Syntactical Query	In this method a syntactical query is done and the arrangement is returned with the recovered files.

This service accomplishes the function of providing recommendations to the student. The first thing it does is to obtain the activities from the environment's semantic annotations and submit a query. This query is classified first as semantic. However, if no result is obtained, a syntactical query can be done. As a result, this web service sends the recovered activity related files.

VIII. VLE Recommender Agent Integration

To integrate this agent with the VLE, it is necessary to use two different technologies. On the one hand, there is the VLE, which is based on PHP language. On the other hand, the agent itself is based on Java.

In order to establish this communication, a SOAP communication protocol is used with a NUSOAP tool in PHP, that is, to enable the sending of messages via the protocol.

Additionally, a VLE entity is created to save the already annotated resources so as not to repeat the resource recommendation to the students.

To activate the two web services, a client was created in the VLE. This client is located in the recommendation module folder; the file is linked to the web services via the following code lines:

```
$client = new
SoapClient("http://localhost:8084/appannot
ador/WSannotador?wsdl");
$client = new
SoapClient("http://localhost:8084/app_agen
t/appAgentWS?wsdl");
```

A. Comments:

In the interface or recommendation module, the student pending activities are described (forums and tasks) under the title "Computer Architecture". When clicking on each activity, the recovered resources are shown.

After the integration, some tests were done with the students in order to verify its precision for recovering relevant files for the student activities.

With the attribute-based technique, it is possible to incorporate information recovery systems. The agent in this case focuses on the most frequently used tests for this type of system. According to [15], these are called precision tests.

IX. Recommender agent assessment

In order to evaluate the VLE recommender agent function, two types of tests were performed: precision and usability. These kinds of tests were applied to two VLE courses. The first one was applied to determine whether the recommended files match the student activities. On the other hand, the usability tests helped to validate the recommendation module use. That is to say, we were able to distinguish whether or not this module was user-friendly, or if there were any navigation problems.

A. Precision Tests Results

The precision value is calculated using the following equation:

$$Precision = \frac{RelevantRecoveredfiles}{Recoveredfiles} \quad (1)$$

In the next table, results of document retrieval are shown.

TABLE III
Precision Results Document Retrieval

Course	Activities	Recovered Files	Relevant recovered files (activity related files)	Precision %
Computer architecture	10	8	7	88%
Computing Theory	10	16	14	87%

As one can see from the above table, the activity related resources were the most relevant files and were going to be recommended by the agent. For each activity, the agent recovered several files. In the case of Computing Theory, 16 files are recovered but only 14 are pending activity related. When all the agent recovered files are relevant for the activity, the agent is considered 100% precise. Nevertheless, this is not always the case because of the presence of false positives and noise, that is to say, non-relevant files are presented as important ones. Another problem is silence, for example, when a relevant file is not recommended.

As seen above, the recommender agent precision is between 87% and 88%. This is an acceptable precision result. Moreover, according to [8], it is in the recommended scale.

B. Usability Tests Results

For this test, it was necessary to obtain the participation of a specific sample consisting on one teacher and a group of students. The students interacted with the recommendation interface and after that, they were invited to answer some questions focusing on their opinions concerning the usage and navigation of the interface.

Based on the usability tests, the recommendation module presentation (interface) was improved, making it more user-friendly.

Another change concerned the recovered files. In some subjects, the suggested resources had little relationship with the activity. This problem was solved by improving their description.

X. Conclusions

Focusing on the obtained results, we can conclude that the recommender agent can provide some personalized support to the student in their learning activities- with an acceptable educational resources recovering precision. Moreover, it can be a starting point for agents to construct learning management systems.

Due to the different problems in each recommendation technique, some semantic web tools can be used for solving some of these problems like the "cold start", which is related to user information (new user and new item problems).

An essential requirement for the agent function is the environment that is used to collect information for making the necessary inferences and recovering the files to be recommended. The semantic annotation process is another indispensable aspect for this kind of system since it allows users to explore the agent functions. Otherwise, their usage would be limited.

This recommender agent seeks to satisfy the user needs by supporting learning activities. In addition, it can be applied to reduce the information over-load for the teachers in each online course in the VLE. The agent would do the tutors' work by suggesting the resources that can help the students in their learning.

References

[1] O. Velez-Langs, C. Santos, "Sistemas Recomendadores: Un enfoque desde los algoritmos genéticos." *Revista Industrial*, vol. 9, 2006, pp. 23–31.

[2] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms", in *Proc. of the 10th international conference on World Wide Web, New York, 2001, pp.* 285–295, doi:10.1145/371920.372071

[3] S. Wang, C. Wu, "Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system", in *Expert Systems with Applications, vol. 38, 2011, pp.*10831-10838, doi:10.1016/j.eswa.2011.02.083

[4] R. Burke, "Hybrid Web Recommender System.", in *The Adaptive Web*, vol. 4321, P. Brusilovsky, A. Kobsa, and W. Nejdl Ed., Berlin Heidelberg: Springer-Verlag, 2007, pp. 377-408.

[5] G.Adomavicius, A.Tuzhilin, "Toward the nextgeneration of recommender system: a survey of thestate-of-the-art and possible extensions", *IEEETrans. on Knowledge and Data Engineering, vol. 17, 2005, pp. 734-749.*

[6] K. Ghauth, N. Abdullah,"Learning materials recommendation using good learners, rating and content-based filtering. Education Tech Research Dev", vol. 58, 2010, pp. 711- 727, doi: 10.1007/s11423-010-9155-4.

[7] I. Pascual-nieto, O. Santos, D. Perez-marin, J. Boticario, "Extending Computer Assisted Assessment systems with Natural Language Processing, User Modeling, and Recommendations based on Human Computer Interaction and Data Mining", in *International Joint Conference on Artificial Intelligence,* Barcelona, 2010, pp. 2519–2524.

[8] O. Santos, J. Boticario, "Users' Experience with a Recommender System in an Open Source Standard-Based Learning Management System". *Lecture Notes in Computer Science,* vol. 5298, 2008, pp. 185–204, doi:10.1007/978-3-540-89350-9_14

[9] M. Khribi,M. Jemni,O. Nasraoui,"Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval", in *Educational Technology & Society*, vol. 12, 2009, pp. 30–42.

[10] H. Drachsler, H. Hummel, R. Koper,"Personal recommender systems for learners in lifelong learning: requirements, techniques and model", in *International Journal of Learning Technology*, vol. 3, 2008, pp. 404–423.

[11] M. Pazzani, D. Billsus,"Content-Based Recommendation Systems",in *The Adaptive Web,* P. Brusilovsky, A. Kobsa, & W. Nejdl, Ed., 2007, pp. 325–341.

[12] Z. Huang, X. Lu, H. Duan,"Context-aware recommendation using rough set model and collaborative filtering",in *Artificial Intelligence Review*, vol. 351, 2011, pp. 85–99.

[13] W. Chen, Z. Niu, X. Zhao,Y. Li,"A hybrid recommendation algorithm adapted in e-learning environments",in *World Wide Web*, 2012, pp. 1–14, doi:10.1007/s11280-012-0187-z.

[14] S. Rusell, P. Norving,"*Inteligencia Artificial Un Enfoque Moderno*", 2nd ed., D. Fayerman, Ed, Madrid: Pearson, 2004.

[15] D. Llidó,"Extracción y Recuperación de Información Temporal",PhD diss. Universitat Jaume I, 2004.