Users' Experience with a Recommender System in an Open Source Standard-Based Learning Management System

Olga C. Santos and Jesus G. Boticario

aDeNu Research Group, Artificial Intelligence Dept. Computer Science School,
UNED C/Juan del Rosal, 16. Madrid 28040, Spain
{ocsantos,jgb}@dia.uned.es
http://adenu.ia.uned.es

Abstract. The paper describes a model for recommendations in learning scenarios which has been designed from empirical findings following usability and accessibility criteria. This model supports course designers in describing recommendations and presents additional information to the user to explain why the recommendation has been provided. A prototype of a recommender system based on this model has been integrated in an open source standard-based learning management system. The main goal of the recommender is to improve the learning efficiency. Examples of recommendations defined with this model are provided. Moreover, a users' experience is reported.

Keywords: Recommendations, eLearning, usability, accessibility.

1 Introduction

Recommender systems (RS) can be applied to many areas where users are to be supported in their decision-making while interacting with large information spaces. They support users in finding their way through the possibilities offered in web-based settings by pre-selecting information a user might be interested in. Recommender technology has traditionally focused on e-commerce activities to select and suggest extra potential purchase to users/consumers, trying to ease the information search and the decision process. Another area where this support is very much demanded is in the eLearning field, where it would be desirable that learners are offered the most appropriate activities and resources to achieve their individual learning goals and support their needs in the most efficient way. Traditional approaches to computer-based instruction have followed a basic strategy to support learning in terms of objectives and learning resources, along with assessments focused on the measurement of learners' performance [1].

Some works have suggested applying recommendation strategies to learning scenarios. In this context, RS should help and support both learners and tutors during the course execution [2]. Learners should be supported in the performance of the course tasks by i) avoiding blockages, ii) improving the performance of the learning process by facilitating the most appropriate course contents and learning paths adapted to the

A. Holzinger (Ed.): USAB 2008, LNCS 5298, pp. 185–204, 2008.

learner's needs, and iii) promoting collaboration among peers. Tutors should be supported i) in the design of ad-hoc recommendations that can be delivered to the learner in the appropriate moment, and ii) in the follow-up of the learners' work by being alerted of troublesome situations [3].

Learning scenarios share the same objective as recommenders for e-commerce applications (i.e. helping users to select the most appropriate item from a large information pool) but have some particularities that have to be taken into account [4, 5, 3]: 1) the requirements (recommendations should be pedagogically guided and not only by learners' preferences, and accessibility barriers should be overcome by considering the user preferences and device capabilities), 2) the user predisposition (learners are not so motivated to continuously provide explicit ratings for each item they access as in e-commerce systems, but in turn they are used to fill in advance information requested by the institution), and 3) the structural context (educational specifications allow to situate the learner in the course). Moreover, the approach here is of lower granularity. We do not intend to recommend a course from a list of available courses regarding the users' preferences –as done in typical recommending systems, where movies or songs are recommended to a user–, but to recommend actions to the learner while performing the activities designed for a given course.

From our experience in aLFanet project (IST-2001-33288), we came to the conclusion that eLearning scenarios should combine design and runtime adaptations to better support users in the full life cycle of the learning process [6]. Following these ideas, we have designed a model to manage recommendations at design time to support the runtime operation. In this way, modeling at design time provides the needed scaffolding to offer recommendations that take into account and dynamic support required for each user.

This model allows defining different types of recommendations, which are available actions in a learning management system (LMS) to be done by learners and tutors. To apply them, different types of conditions are defined at design time, which are later computed at runtime against the current context (user, course and device). Although this model has been designed based on the requirements of learning scenarios –no matter the pedagogy applied– the ideas presented in this model can be reused in other domains.

In this paper, first we describe the model, present how usability and accessibility criteria have been considered in the user interface and comment on the sources for the data gathering. Next we present some types of recommendations that can be provided by typical LMS and show examples of recommendations defined with that model. A prototype of the system has been integrated in an open source standard-based LMS called dotLRN, which is presented in section 4. Afterwards, we describe the results of an experience with users, where learners and tutors were given a subset of recommendations in a course run in dotLRN platform and asked for their feedback. Finally, we comment on some pedagogical aspects of our approach and end-up with some concluding remarks and future works.

2 The Recommendations Model

Works in lifelong learning scenarios show that recommendations in them should be based "on most relevant information about the individual learner and the available activity, history information about similar learners and activities (learning path), guided by educational rules and learning strategies, aimed at the acquisition of learning goals" [5]. Moreover, other works show that accessibility requirements and device capabilities have also to be taken into account [7].

To support the definition of recommendations, we have worked out a model where a set of elements have to be defined to facilitate the runtime process. This model has been defined based on empirical findings and covers the following objectives: 1) supporting the course designer in describing recommendations in learning inclusive scenarios, 2) presenting additional information to the user to explain why the recommendation has been offered, and 3) requesting explicit feedback from the user when she has shown interest in the recommendation process to improve the recommender [3]. The followed methodology includes brainstorming sessions with psycho-pedagogical experts and evaluation experiences with end-uses, as the one reported in section 5.

2.1 Elements of the Model

The elements proposed for the model are: the categories, the techniques, the origin, the explanation, the timeout restrictions and the conditions. The later relates to the context information. The interrelations are shown in Fig. 1.

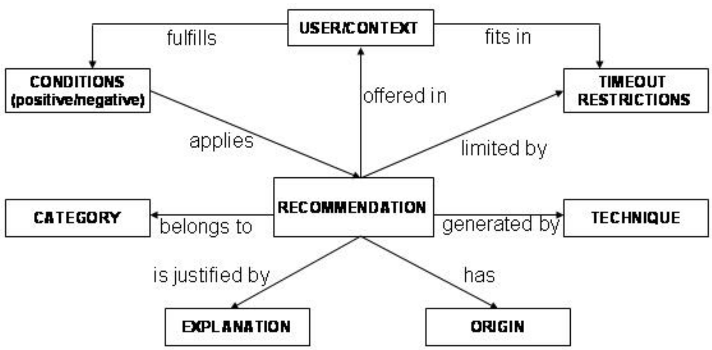

Fig. 1. A model for recommendations in learning scenarios

The Figure 1 summarizes the modeling options that characterize a recommendation. A recommendation belongs to one of the eight categories defined and can be generated by a single technique –or a combination of– techniques. At design time, the course designer selects the category to which the recommendation applies, defines the conditions and timeout restrictions and adds an explanation. The structure of the recommendation can point to the available recommendation types, which are services in the LMS. The origin and technique are dependent on the way the recommendations are generated. At runtime, the conditions and restrictions are checked against the user model and the context at hand. If applicable, the recommendation is offered to the user in an explicative, usable and accessible user interface (see below).

2.1.1 The Category

The category element classifies the scope of the recommendations offered. Our experience shows that depending on the course situation some categories (scopes) of recommendations will be more useful than others to help the user carry out her task in a more efficient, effective and satisfactory way. In particular, two states have been defined. First, the transitory state, when the user is new to the system and the course. Second, the permanent state, when the user is used to both platform and course methodology [8]. Furthermore, this information can be used to compare the performance of different RS with an ideal behavior by defining the ideal Kiviat figure for each situation. We have identified the following eight categories:

- Motivation (Mo): provides messages to motivate the learner when working in the course so she does not get frustrated if the results are lower than expected or if she requires a lot of time to carry out the given tasks.
- Learning styles (LS): suggests a way of learning the contents or the appropriate alternative content which applies best to the user preferred way of learning.
- Technical support (TS): provides hints for using the LMS functionalities or the browser.
- Previous knowledge (PK): takes into account what the user already knows, so some parts of the course are given more emphasis (if the user has not previous knowledge on them).
- Collaboration (Cl): fosters sharing contributions, communicating with course members, given the opinion on the peers work, etc.
- Interest (In): recommendations focus on those issues that the user has interest.
- Accessibility (Ac): deals with accessibility issues, such as recommending an alternative format that matches the user accessibility preferences.
- Scrutability (Sc): promotes self-reflection by telling the user what the system knows about her.

The above list of possible values for the categories is open to changes if the results from evaluations show that there are some overlaps among categories, or missing values are identified.

2.1.2 The Technique

This element refers to the recommendation technique used to generate the recommendations. According to [5], memory-based recommendation techniques are the most appropriate. Following that approach, we propose to combine the most appropriate for each situation. We consider the following techniques:

- Matching conditions (MC): selecting suitable recommendations from those defined ad-hoc by the professor in terms of conditions that apply to the current context.
- User-based collaborative filtering (UbCF): users that rated the same item similarly probably have the same taste and thus, can be recommended similar actions.
- Item-based collaborative filtering (IbCF): items rated similarly by users are probably similar, and thus, can be recommended for users who liked related items.
- Demographic collaborative filtering (DCF): users with similar attributes are matched, and similar actions recommended to them
- Ratings-Attributes mix (RAm): positive rated items by learners are recommended to similar learners.

- Case-based reasoning (CbR): if a user likes a certain item, she will probably like similar items in terms of the attributes they own.
- Attribute-based rules (AbR): item attributes are match to the preferences stored in the user profile.

In this way, social-based filtering (specially user-based and item-based techniques) are combined with information-based filtering. If the former can be applied, learners benefit from the experience of others.

2.1.3 The Origin

The origin identifies the source that originated the recommendation that has been given to the learner. We have defined the following four in order to motivate the learner to follow them:

- Preferred (Pf): it matches the user preferences.
- Popular (Pp): similar users have already found useful that recommendation.
- Tutor (Tu): the recommendation has been added by the tutor of the course.
- Course design (CD): the recommendation was specified in the course design.

Evaluations with users may result in new values for the origin element.

2.1.4 High Level Explanation

A high-level explanation of the recommendation is given for each recommendation, which explains to the user in detail why the recommendation was offered to her. This (as many of the above elements) is intended to promote trust in the user, as commented later.

2.1.5 Timeout Restrictions

Timeout restrictions to inform about the validity of the recommendation, and can be defined by an absolute or relative date, or a certain condition that can take place.

2.1.6 Applicability Conditions

Conditions for offering the recommendations define what values should take place for a user and her context at runtime to be given the corresponding recommendation. This conditions can be specified by identifying the set of values that should take place to be given the recommendation (positive conditions) and/or the set of values that should not take place to be given the recommendation (negative conditions). These values consider the user model attributes and the context.

Regarding the user information, the following values can be considered:

- A particular user (Us). Usually, recommendations are defined generic to any user that matches the required condition, but it may happen that a certain recommendation is to be given to a particular learner.
- Similar to another user (SU). Comparison on the user model information among the given user and the user using being recommended will be done.
- The role the user has in that moment (Ro). It can be a member of a community, the professor or a learner in a course.
- Learning styles of the user (LS). The recommendation may be appropriate for users with strong values in a particular learning style dimension.

- Technology level (TL). Depending on the technology level, a recommendation can be suitable for a user or not.
- Collaboration level (CL). The collaboration level is very relevant in virtual settings to assure the knowledge and experiences are interchanged among members.
- Accessibility preferences (AP). The accessibility preferences are critical in inclusive scenarios and can be considered in relation to the content of the recommendation or the interactions taken place in the course.
- Interaction data (ID). It considers the data regarding the interaction style, state and actions of the user
- Knowledge level (KL). It takes into account the knowledge level the user has regarding the learning objective (or competence) she is working in the current context.
- Interest level (IL). It takes into account the interest level the user has regarding the learning objective (or competence) she is working in the current context.

In turn, for the context information, the values are as follows:

- Platform (Pl). Recommendations are given to the learner when she is using the LMS facilities and can be applied when the user is working on a specific environment with its given resources. It can also take into account actions (i.e. read, write, create, etc.) done on objects (e.g. forum message, file, course assessment, learning object, etc.) of the LMS.
- Device capabilities (DC). Some recommendations may be or not suitable depending on the capabilities of the device being used.
- Instructional design (ID). Provides the situation the user is in the course, especially, the learning objective (or competence) being addressed in the current activity.

When designing a recommendation, not all the above properties should be filled in, only those that should be checked at runtime.

2.2 The User Interface

As in any RS, a critical issue is how to present recommendations to the user. A sensible approach is to offer a subset with the most relevant recommendations for the user, which she has the freedom to follow or not. The information shown for the recommendations and the way it is presented in the user interface can influence the attitude of the user towards the system. Providing an explanation on how the recommendation has been produced increase the user trust of the system [9].

In this section, first, we discuss the structure defined to present a recommendation on the screen. Next, we comment on the usability and accessibility criteria followed.

2.2.1 Structure of the Recommendation

The information presented to the user consists in a list of one or more recommendations. The recommendation list consists of an introductory text for the user (called 'greetings') plus a list of suggestions of actions to do by the user. Each of these suggestions (called 'recommendation') is a sentence ('content') describing the suggested action to do by the user, where a part of the sentence may be defined as a hyperlink (or more commonly called, 'link'). Since the link may most of the times be within the suggestion sentence, this (the content) will be divided into two parts, separated by a placeholder which is placed where the link should go. In more detail:

- The content: the sentence that is shown to the user and includes the link (which is placed instead of a placeholder).
- The text: the text shown on the link
- The title: the title attribute of the link
- The pointer: the URI that opens the link, it can be a URL or an object identifier from the LMS, depending on the type.
- The type: it can be internal to the LMS (if the pointer is an object identifier from the LMS) or external (if points to a URL, both from inside or outside the LMS).

The Figure 2 (in section 4) shows how the graphical user interface of the recommending system looks like for four recommendations.

2.2.2 Usability Criteria

Usability relates to the clarity with which the interactions with the RS are designed in order to make it easy and intuitive to use. Studies have shown that in order to evaluate a RS, the user satisfaction is as important as the results obtained from accuracy metrics [10]. Providing good explanations can increase the user's trust on the system's recommendations. In the learning scenario, if the user follows the recommendations (and assuming that they are appropriate) it should be quicker and easier to achieve her learning goals. Therefore, providing good explanations on why recommendations have been provided to the user can improve the user satisfaction, and thus, the quality of the system.

Adapting [11] to the particularities of learning inclusive scenarios, we have designed the user interface to cope with the following aims:

- Transparency: explaining how the system works, that is, why the recommendation have been given to the user. To achieve this, we offer a link to the user profile (see Fig. 2 in section 4) and have created an explanation page (see Fig. 3 in section 4).
- Scrutability: allowing users to tell the system that it is wrong. This functionality is available from the explanation page (see Fig. 3 in section 4).
- Trust: increasing users' confidence in the responses given. For this reason, we try to offer good explanations for the recommendations given (to allow the user understand the system behavior when the system offers a wrong recommendation).
- Effectiveness: helping users achieve the learning goals. To achieve this, recommendations are given to make the user achieve the learning goals.
- Efficiency: helping users achieve the learning goals faster or with fewer resources. In this respect, recommendations are given to make the user achieve the learning goals in a quicker and easier manner.
- Persuasiveness: convincing users to follow the recommendations and do the actions suggested. For this reason, icons are added to identify the origin of the recommendation, so the user can select the preferred one.
- Satisfaction: increasing the ease of usability when interacting with the recommender. In this sense, recommendations appear when they are relevant for the context.

For usability reasons, the number of recommendations to be provided has also been limited. The number will depend on the device capabilities and the user accessibility preferences (such as the screen size and the font size). In any case, the user is given the option to get the full list of available recommendations. Moreover, she can also access to all the recommendation that has followed.

2.2.3 Accessibility

Accessibility deals with designing user interfaces that are flexible to meet different user needs, preferences, and situations. This flexibility makes possible for people with disabilities perceive, understand, navigate, and interact with the Web, but also benefits people without disabilities in certain situations, such as people using a slow Internet connection or driving a car, people with temporary disabilities such as a broken arm, and people with changing abilities due to aging.

Regarding accessibility, the user interface is compliant with the accessibility guidelines from the W3C Web Accessibility Initiative [12]. To support those guidelines the structural information of the recommendation asks for a title description in the link to clearly identify the target of each link (checkpoint 1.3 1 from the Web Content Accessible Guidelines 1.0). Moreover, the icons presented in the user interface to make easier to understand the reasons for the recommendation are described with an alternative text (checkpoint 1.1). Header elements have also been used to convey document structure (checkpoint 3.5) and the lists structure to present the recommendation has been properly marked-up (checkpoint 3.6). The language used has also been written clearly, to facilitate the understanding by deaf from birth and cognitive disabled users.

However, the accessibility level does not depend only on the output from the RS, but also on the way the LMS presents the HTML elements.

2.3 Data Gathering

The data managed by the model is obtained from different sources and at different phases. At design time, the recommendations can be defined in terms of the category, the explanation, the origin, the applicability conditions and the time out restrictions. This information can be filled in by the course administrator via a administration graphical user interface or automatically by intelligent agents that apply some recommendation strategies [7].

At runtime, the user features of the user at hand are obtained from the user model and data from interactions (active and passive) can be obtained from the tracker component of the LMS.

3 Recommendations

Recommendations refer to the different actions that can be recommended, which are available functionalities in the LMS. Currently the system is configured to provide certain types of recommendations (but it can be easily extended for new ones, provided that the proper information is defined). First, we present the type of recommendations available. Next, we present a table of some recommendations defined following the model.

3.1 Types of Recommendations Available

The following recommendations are configured in the system:

- Learning styles (LS): points to the learning style inventory package to compute the learning styles of the user.
- Help (Hl): points to the general help page of the platform or to any of its subsections that is relevant to the current user context (contextual help)
- Post message (PM): points to a particular message of the forum and tells the user to provide a response to it.
- Read message (RM): points to a particular message of the forum and tells the user to read it.
- Upload link (UL): points to a particular folder in the file storage and tells the user to create a link there. The link can be internal to the platform or a external URL.
- Upload file (UF): points to a particular folder in the file storage and tells the user to upload a file to it.
- Read file (RF): points to a particular file and tells the user to read it.
- Read FAQ (RQ): points to a section in the FAQ and tells the user to read it.
- Fill assessment (FA): points to a particular assessment and tells the user to fill it in.
- Post a message in a blog (PB): points to a blog and tells the user to write a post.
- Comment a blog message (CB): points to a message of a blog and tells the user to comment it.
- Participate in chat (PC): points to a chat room and tells the user to participate.
- Make rating (MR): suggest to explicitly rate some elements of the environment, (the pointer will be to the object to be rated, e.g. a file), the collaboration level of a user, the difficulty level of an activity, the relevance of an activity, etc.
- Make comment (MC): suggest to comment a platform object, including items from the instructional design (the pointer will be to the object to be rated, e.g. a file)
- Read comment (RC): suggest to read the comment done on an object by a user.
- Read external link (RE): points to an external URL and tells the user to read it
- Enter a course (EC): points to a course space and tells the user to enter.
- Do activity (DA): points to an activity and tells the user to do it.
- Read resource (RR): points to a resource and tells the user to read it.
- Work on objective (WO): suggests focusing the work on a particular learning objective (or competence), whose description is given in the link.
- Online classmates (OC): shows the users from the same course currently connected and recommends a synchronous communication.
- See user model (SU): suggests seeing information inferred about the user from her interactions in the system.
- See user statistics (SS): suggest to see the statistics of the user in the system (or another user –learner– in the case of the professor)
- Alert deadlines (AD): reminds a close deadline and points to the corresponding instructions.
- Accessibility features (AF): provides advice on using the accessibility features of the browser and the platform.
- Follow-up user (FU): suggests following the contributions done by one of the learners.

- Plain text (PT): shows plain text to the user, there is no link. Mainly to be used to offer motivational messages.

New types of recommendations can be easily added to the system, provided that the corresponding functionality is available in the LMS. In order to manage this, we have proposed an administration user interface that is currently being implemented.

3.2 Instances of Recommendations

To validate the model, we have compiled instances of recommendations from instructional design experts and mapped them into the model proposed. Some of the samples gathered are mapped in the following table. The technique used and the origin will be assigned at runtime, when the recommendations are produced.

Table 1. Description of recommendations according to the model (acronyms are defined in the corresponding sections)

Recommendations	**Type**	**Cat.**	**Conditions (*)**		**TO restrictions**	**Orig.**
			User	**Context**		
if Learner.inPlatform_tool=X & Learner.technology_level=low → link to help.toolX	Hl	TS	+ID(low number of sessions), +TL(low)	+Pl(action on tool X)	Until clicked OR up to 5 sessions	Pp
if Learner.averageTimePerSession = low → "Sure you have sometime today to stay a bit longer in course!"	PT	Mo	+ID(low average time per session)	+ID(activity.difficulty Level=high)	Average time per session increases 20%	Tu
if Learner.LearningStyle=global → Show first the contents table	RR	LS	+LS(global)	+Pl(course space) +ID(course)	Table content seen	CD
if Learner.knowledge_level=low → Give additional material	RR	PK	+KL(low in objective)	+Pl(course space) +ID(course)	Resources accessed OR knowledge_level increased	CD
if Learner.PreferredFormat=auditive → course presentation in audio	RR	Ac	+AF(format auditive)	+Pl(course space) +ID(resource has alternative)	Resource accessed	Pf
if LearnerA.user_model ~ LearnerB.user_model → tell LearnerA to evaluate LearnerB.contributions	MR	Cl	+SU (user B)		Ratings done by learner A	Pp
if LearnerA.rating.inObject1=0 & LearnerB.rating.inObject1=5 → tell Learner.A & LeanerB to justify rating in thread	PM	Cl		+UserA.Pl(low rating in object1) +UserA.ID(courseA) AND +UserB.Pl(high rating in object1) +UserB.ID(courseA)	Message posted	Pp
if Learner.interest.objectiveA=high→ give items on objectiveA	WO	In	+IL(high in objectiveA)	+Pl (course space) +ID(courseA)	Next activity available	Pf
if Learner.participation=high → Show user model	SU	Sc	+ID(high participation)	-Pl(user model accessed)	User model seen	Pp
if Tutor inCourse.A → follow-up learners low interactive level in courseA	FU	TS	+Ro(tutor)	+Pl (course space) +ID(courseA)	Clicked on learner info	CD

(*) A "+" means a positive condition on that feature; a "-" means a negative condition on that feature (see section 2.1.6).

For instance, the first recommendation from the table will show the help page of the forum tool to the current user in the LMS if she has used the system less than five times, her technology level is low and she has not read the forum help yet.

4 Integration in dotLRN Learning Management System

Current developments have focused on providing the infrastructure for the RS to allow offering recommendations in the LMS user interface. In particular, an open source infrastructure for open standard-based LMS has been implemented to enrich LMS functionality with a dynamic support based on users' interactions. This initial prototype has been integrated in dotLRN [13]. The following snapshots present the user interface of the recommending system.

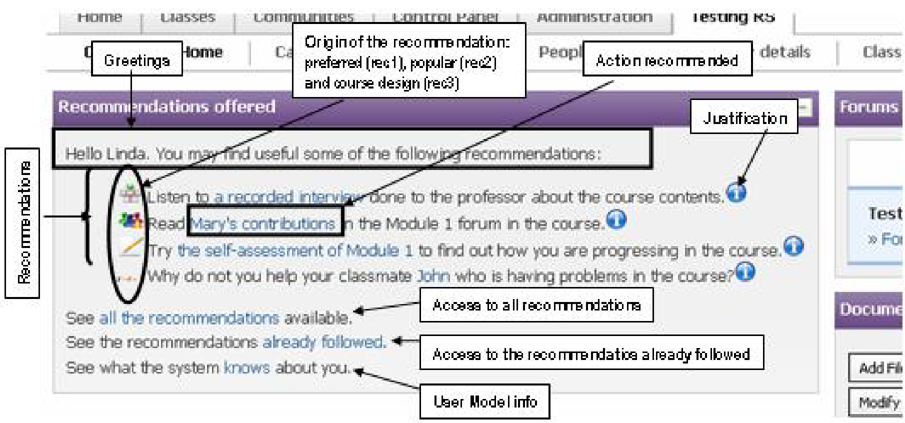

Fig. 2. Snapshot of a recommendation presented in the LMS

The icons from Fig. 2 represent the origin of the recommendation and are used to entice the user to follow the recommendations. Thus, depending on her profile and experience, she may be more confident to follow recommendations either useful for similar learners or those provided by the tutor.

The page with the explanation of the recommendations (Fig. 3) intends to get explicit feedback from the user (so the users can tell the system if it is wrong). If the user has clicked for getting more details on the recommendation, we can assume that she can find reasonable to be asked for her opinion. Therefore, she is presented with two links to close the page, one for the case that she has found useful the recommendation, and the other one for the opposite case. There is another link to provide more detailed feedback. It was found that users are ready to provide more input to the system in order to receive more relevant recommendations [9].

To avoid to overly clutter the interface and overwhelm the user with information, the number of recommendations has been limited, but the user is given the opportunity to access all the recommendation that were followed in the past as well as the

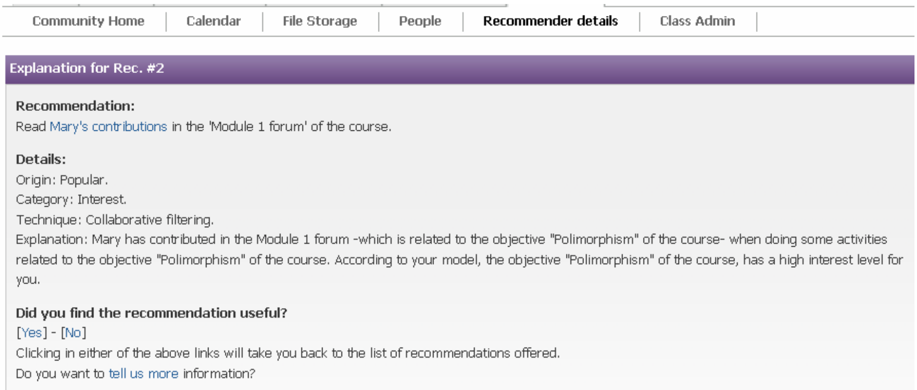

Fig. 3. Explanation for the Recommendation #2

whole list of recommendations available. Giving access to the user profile and the explanation for the recommendation offers transparency on the recommender output and system logic and therefore participate in increasing trust [9].

5 The Users' Experience

At this point, we describe the results of an experience ran during a course on Accessibility and Information and Communication Technologies, organized by aDeNu research group in July 2008 as part of the UNED offer of summer courses[1]. The focus of the experience was not put on whether the users found useful the recommendations given, but if the way recommendations were defined and presented to the users was useful and understandable for them. The set of recommendations offered was relevant to show the users how the recommending system would behave, and that this behavior would differ for each user, depending on each user's individual needs and preferences.

Fourteen users took part in the experience, although only nine of them reported valid results. One of them was visual impaired and another one was physically impaired. The other seven had no declared disability, but had worked with disabled users and were aware of their difficulties when using web-based environments.

The users were given a subset of recommendations (from those types described in section 3.1) and asked for their feedback. After an hour using the system, they were given a questionnaire. This questionnaire addresses functionality, usability and accessibility issues of the recommending system. Now we comment on the most relevant questions.

Most of the students considered that having a RS integrated in the LMS is very useful.

---

[1] Session on 'Future perspectives: towards a University fully accessible. Experiences in research projects of UNED (II)': http://apliweb.uned.es/cverano/cursos.asp?idcurso=126.

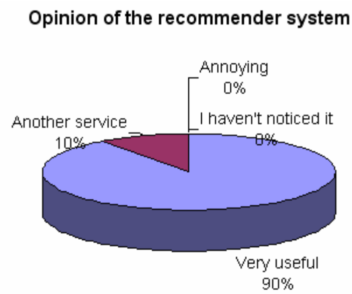

Fig. 4. Opinion of the RS by the users

Users were presented a list of recommendations (see below) to select the types they would prefer. From that list, seven types of recommendations were selected by more than half of the students: 1) "Fill in a learning styles questionnaire, so the system can be adapted to me", 2) "Read some section of the help, if there is a service in the platform that I don't know", 3) "Get alerts on deadlines to hand in an activity", 4) "Read a message in the forum that has information that may be relevant to me", 5) "Students that are on line, to set up an asynchronous connection", 6) "Advice to take advantage of the accessibility features of the platform and the browser" and 7) "Read a file uploaded by the professor or a classmate".

In turn, there were three recommendations that were selected by less than a quarter of the students: 1) "Rate some contribution done by a learner", 2) "Fill in a self-assessment questionnaire" and 3) "Access an external link of the platform".

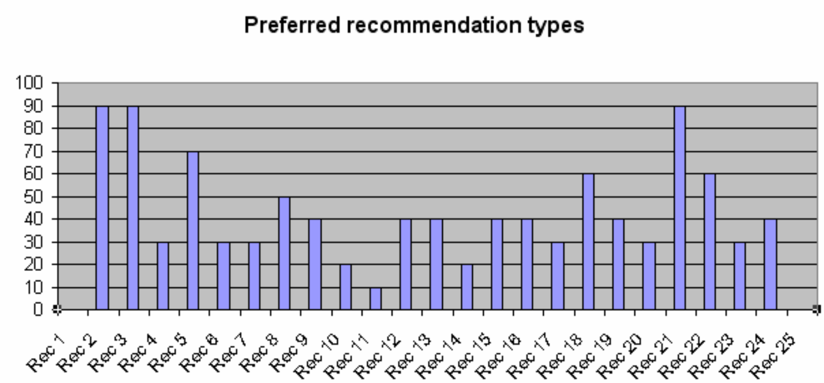

Fig. 5. Preferred recommendation types for the users

The recommendations offered in the experience were not computed by the recommending system itself, but defined ad-hoc for the experience (applying recommendation strategies to compute them is an on-going development work). However, we were interested in getting their feedback on which types of recommendations they would expect more likely, in order to guide our development efforts. The list of recommendations given to the users to rate for relevance is the following:

- Rec 1: None (I decide what to do)
- Rec 2: Fill in a learning styles questionnaire, so the system can be adapted to me
- Rec 3: Read some section of the help, if there is a service in the platform that I don't know
- Rec 4: Put a message on the forum to ask or share information about the learning objective I am working on the course
- Rec 5: Read a message in the forum that has information that may be relevant to me
- Rec 6: Upload a file with my contributions, that can be useful for others (e.g. the professor to evaluate it)
- Rec 7: Upload a link that can be useful to my classmates
- Rec 8: Read a file uploaded by the professor or a classmate
- Rec 9: Read a section of the FAQ
- Rec 10: Fill in a self-assessement questionnaire
- Rec 11: Rate some contribution done by a learner
- Rec 12: Make a comment to some contribution
- Rec 13: Read some comment of a classmate
- Rec 14: Access an external link of the platform
- Rec 15: Carry out some particular task from the course design
- Rec 16: Read a specific material from the course design
- Rec 17: Work on a specific learning objective of the course
- Rec 18: Students that are on line, to set up an asynchronous connection
- Rec 19: Access my user model, to see what the system knows about me
- Rec 20: See usage statistics from the platform
- Rec 21: Get alerts on deadlines to hand in an activity
- Rec 22: Advice to take advantage of the accessibility features of the platform and the browser
- Rec 23: Messages without any action (e.g. motivational messages)
- Rec 24: Follow-up of a classmate contributions
- Rec 25: Other

Results showed that the learning style information is considered critical, and they agree to fill in a learning style questionnaire in order to have the system adapted to it. Since the recommender is applied to a course, and courses are established by milestone actions that users have to accomplish, learners liked very much the idea of getting reminders of activities deadlines. Being aware of information added to the platform is also relevant for them (messages or files uploaded). They are also willing to establish synchronous contact with on-line classmates. Advice to take advantage of the accessibility features of the platform and the browser is also welcome. However, they gave less importance to evaluating other learners' contributions and being recommended to fill in assessments. Both are quite relevant to get feedback from the user to improve and evaluate the performance of the RS, since it provides very useful information, especially to check the validity of a property learnt by the system from the users' actions (which the system considers as implicit ratings). However, this has to be pedagogically driven (e.g., the learning design considers useful to let the learner assess a particular learning activity when she performs badly) since the RS follows a non-intrusive policy. Our view is that users did not like this type of recommendations thinking on being intrusively asked for feedback. Finally, accessing external links is not considered very relevant, either.

With respect to usability, a majority agreed that icons were clear. However, it has to be noticed that 20% of them had not paid attention to them.

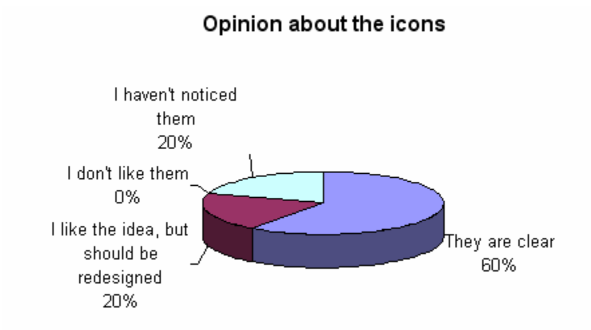

Fig. 6. Opinion about the icons in the user interface

They were also asked what kind of information they would like to receive about the recommendation. The four information sets given was found relevant for the majority of users. Especially, the high level explanation of the recommendation.

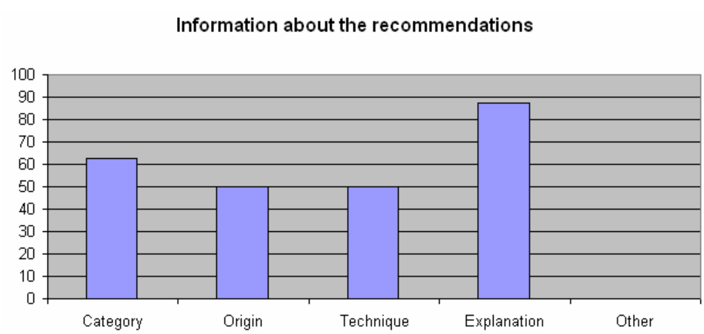

Fig. 7. Information to be given about the recommendations

Students were asked for their preferred category, among those defined. The most relevant for them was learning styles, which is consistent with the fact that being recommended to fill in the learning styles questionnaires to obtain more adapted recommendations was the most relevant for them. The collaboration category is selected as the less relevant. This is also consistent with the fact that not many of the recommendation types associated to collaboration had been highly selected.

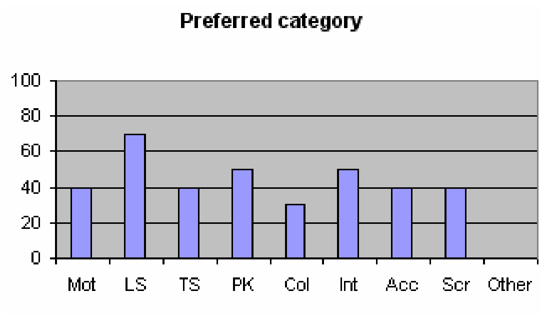

Fig. 8. Preferred categories for the students

Finally, users were asked if they had found any accessibility problem during the experiment. None of them reported any.

6 Pedagogical Considerations

In the previous sections we have presented a model to design recommendations for inclusive learning scenarios. The recommendations cover different scopes (categories) and our more than nine years in teaching on-line courses (and mainly teaching instructors how to teach on-line courses) has shown us that different support is required in the different situations of the course. Moreover, a good design done in advance – which prepares the hooks for the dynamic adaptations– improves the quality of the learning experience [6].

From a small scale experiment, we have detected that those recommendations based on the users' learning styles are highly relevant for them. Literature shows that there may be a direct relationship between learning styles and cognitive traits [14]. Based on that we could extend our recommendations model to cover cognitive traits such as working memory capacity, inductive reasoning ability and associative learning skills from the Cognitive Trait Model [15], as well as related information (field-dependency and thinking style). However, the modeling granularity and the corresponding recommendations should be manageable from the designer viewpoint.

Another issue to consider with respect to the recommendation model is how to deal with dynamic media. Recent studies show that dynamic media can support learning when limited cognitive resources, cognitive load and learners' mental representations are taken into account during the design and development of learning material. By managing recommendations, dynamic media can be tuned to the learners' experience, expertise and previous knowledge [16].

The design process for a recommendation should consider similar steps to those defined to develop a reusable, platform independent, objective based and adaptive course: 1) creation of the course material, 2) annotation with metadata, 3) define the instructional design to be applied guided by learning objectives, and 4) build the adaptive scenario that consider the runtime environment and allow delivering the adaptations to the individual learner needs [17].

Studies show that the key for designing interfaces for eLearning inclusive settings is the concept of 'ease of use', which requires the focus on the end users to address usability aspects [18]. This keeps with the methodology followed in our experiments, which is user-oriented and involves users in the very early stages of the design and development process, the so-called constructive technology assessment [19].

Cultural factors also influence the look and feel of interactive systems and every single individual develops a specific culture in terms of characteristics, behaviors, attitudes and values that affect all levels of human computer interaction (surface, functionality and interaction) [20]. If a system knows the cultural preferences of the end-user, it can adapt to them to reduce the mental workload, prevent mental distress and increase expected conformity [16].

7 Concluding Remarks

A RS in a learning inclusive scenario should provide the LMS with a set of recommendations personalized for the current user (learner or tutor) and context (i.e. the situation in the course –learning objective been worked– and the capabilities of the device used) to help the user select the most appropriate task in order to improve her learning efficiency. Thus, they require a particularized approach from those applied to e-commerce services.

We have developed a model of recommendations that considers the type of recommendations, the technique used, the origin that produced the recommendation, the category that the recommendation belongs to, the conditions that should take place at runtime to offer the recommendation, the timeout restrictions and a high-level explanation for the user justifying why the recommendation was produced.

Recommendations have been modeled according to evaluations with experts and users and are intended to facilitate the dynamic and inclusive support to learners during course execution. This model has been designed to i) support the course designer in describing recommendations in learning inclusive scenarios, ii) present additional information to the user to explain why the recommendation has been offered, and iii) request explicit feedback from the user when she has shown interest in the recommendation process to improve the recommender.

We have developed a user interface to present the recommendations in the LMS, which has been designed to be accessible, usable and explicative. In this sense, we have defined two levels of information given to the user. The first level is the information shown in Fig. 2. The user is given the list of recommendations and she can directly follow any of them. However, if the user would like to know more about the recommendation process, she is provided with more links where details are given. If she clicks in any of them, she is explicitly showing some interest on the recommendation process, and thus, it may be likely that she is receptive to give us explicit feedback on the recommendation process. For this reason, in the explanation window (Fig. 3) we explicitly ask her for feedback that is very helpful to improve the performance of the learning system.

A prototype has been tested with fourteen students from a summer course at our university. The number of users that have participated in the evaluations is small since the required developments are still on-going and not yet ready to carry out large scale experiments, so the results cannot be considered as concluding. But they provide valuable feedback that is useful for the next development phases of the system. In fact, we are applying an agile development methodology that involves users from the early stages of development, allowing changing the direction of the developments if outcomes from evaluations are not as expected. We are currently tuning the system before facing large-scale evaluations, including all the above issues identified such as the pedagogy support and graphical user interface.

8 Future Works

Our next step regarding the user interface is to embed the recommendations within the LMS services: instead of having all the recommendations clustered together within one module (or portlet), recommendations would be spread out and displayed in the most relevant locations, specially the modules that structure the learning design of the course. The objective is to offer highly contextual recommendations to the user, in a way of a contextual help, making them more obvious and relevant and therefore hopefully more efficient. Moreover, we have proposed a graphical user interface to administer the recommendations model, which is currently under development. This interface will be useful to prepare the following experimental settings since it will allow non technical instructors manage the recommendations.

The RS presented here is being integrated with the developments of ADAPTAPlan [21] and EU4ALL [22] projects. ADAPTAPlan provides a multi-agent architecture that offers adaptation based on three different user characteristics: i) competences, ii) learning styles and iii) context. EU4ALL is working on developing an open service architecture for accessible lifelong learning that accommodating the diversity of ways people interact with technology and the content and services it delivers taking into account the individual user needs. Moreover, these developments will be used to run experiences in other projects, such as ‘Accessibility for All in digital alphabetization’ (PAV-020000-2007-171) where accessible standard-based courses designed following the ALPE methodology [23] are offered to reduce the digital gap of all, especially people with disabilities and elderly people.

Another field to explore is to best adapt the RS to the new evolution of LMS, inspired from the Web2.0 trend, which leads toward more user collaboration, social interactions and user generated content. The flexibility required to address this developments is available through the use of web services technology.

In the meantime, we are also working on the prototype, to implement the recommendation techniques proposed in the model and facilitate the automation of data gathering to be able to run experiments with a larger number of users.

To increase coverage results additional evaluations will take place during the Fall term in several courses at the Computer Science School and under the program for ongoing education at our University.

Acknowledgement

Authors would like to thank the European Commission and the Spanish Government, for funding the research projects of aDeNu group, as well as the experts and users that have participated in the experience reported. Specially mentions are to be given to Carmen Barrera for her detailed comments on the recommendations modeling, Emmanuelle Raffenne for the suggestions for the design of the database model and for the implementation of the client side in OpenACS/dotLRN, Jorge Granado for the support in the development of the RS web service interfaces, Héctor Romojaro for setting up the required environments in the server, Emmanuelle Gutiérrez y Restrepo for her useful comments on the users' perception of the system and for presenting the experience to the participants of the summer course, María José Aguilar for her suggestions regarding the user interface, and Olga Revilla and Ludivine Martin for reviewing the paper. In addition, we highly appreciated the suggestions made by the anonymous reviewers.

References

1. Day, R., Payne, L.: Computer managed instruction: An alternative teaching strategy. Journal of Nursing Education 26(1), 30–36 (1987)
2. Zaine, O.R.: Building a Recommender Agent for e-Learning Systems. In: Proceedings of International. Conference on Computers in Education, Auckland, New Zealand, December 3-6 , pp. 55–59 (2002)
3. Santos, O.C., Boticario, J.G.: Modelling recommendarions for lifelong learning. In: Proceedings of the workshop on RecommenderSystems and Personalized Retrieval at ICADIWT 2008 (in press, 2008)
4. Tang, T., McCalla, G.: Smart Recommendation for an Evolving E-Learning System. In: Workshop on Technologies for Electronic Documents for Supporting Learning, International Conference on Artificial Intelligence in Education (2003)
5. Drachsler, H., Hummel, H.G.K., Koper, R.: Personal RS for learners in lifelong learning: requirements, techniques and model. International Journal of Learning Technology (in press, 2007)
6. Boticario, J.G., Santos, O.C.: An open IMS-based user modelling approach for developing adaptive learning management systems. Journal of Interactive Media in Education, JIME (September 2007)
7. Santos, O.C., Boticario, J.G.: Improving learners satisfaction in specification-based scenarios with dynamic inclusive support. In: Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies (ICALT 2008), pp. 491–495 (2008)
8. Santos, O.C.: Recommending in inclusive lifelong learning scenarios: identifying and managing runtime situations. In: Proceedings of the WI-IAT Doctoral Workshop at Web Intelligence and Intelligent Agent Technogoly conference (in press, 2008)
9. Swearingen, K., Sinha, R.: Beyond Algorithms: An HCI Perspective on RS. In: ACM SIGIR 2001 Workshop on RS (2001)
10. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: How accuracy metrics have hurt RS. In: Extended Abstracts of the 2006 ACM Conference on Human Factors in Computing Systems (2006)

11. Tintarev, N., Masthoff, J.: A Survey of Explanations in RS. In: Uchyigit, G. (ed.) Workshop on RS and Intelligent User Interfaces associated with ICDE 2007, Istanbul, Turkey (2007)
12. W3C. Web Accessibility Initiative. Retrieved on May 23 (2008), `http://www.w3.org/WAI/`
13. Santos, O.C., Raffenne, E., Granado, J., Boticario, J.G.: Dynamic support in OpenACS/dotLRN: Technological infrastructure for providing dynamic recommendations for all in open and standard-based LMS. In: Proceedings of the International Conference and Workshops on Community based environments (2008)
14. Graf, S., Lin, T., Kinshuk, J.: The relationship between learning styles and cognitive traits - Getting additional information for improving student modelling. Computers in Human Behavior 24(2), 122–137 (2008)
15. Lin, T., Kinshuk, P.A.: Cognitive trait model – a supplement to performance based student models. In: Proceedings of international conference on computers in education, pp. 629–632 (2003)
16. Holzinger, A., Kickmeier-Rust, M., Albert, D.: Dynamic Media in Computer Science Education; Content Complexity and Learning Performance: Is Less More? Educational Technology & Society 11(1), 279–290 (2008)
17. Santos, O.C., Boticario, J.G.: Meaningful pedagogy via covering the entire life cycle of adaptive eLearning in terms of a pervasive use of educational standards: The aLFanet experience. In: Innovative Approaches for Learning and Knowledge Sharing, Proceedings, pp. 691–696. Springer, Berlin (2006)
18. Ebner, M., Holzinger, A., Maurer, H.: Web 2.0 Technology: Future Interfaces for Technology Enhanced Learning? In: Stephanidis, C. (ed.) HCI 2007. LNCS, vol. 4556, pp. 559–568. Springer, Heidelberg (2007)
19. Heimgärtner, R., Holzinger, A., Adams, R.: From Cultural to Individual Adaptive End-User Interfaces: Helping People with Special Needs. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2008. LNCS, vol. 5105, pp. 82–89. Springer, Heidelberg (2008)
20. Rip, A., Misa, T.J.: y Schot, J. Managing Technology in Society. The approach of Constructive Technology Assessment, London, Pinter (1995)
21. Baldiris, S., Santos, O.C., Huerva, D.J., Fabregat, R., Boticario, J.G.: Multidimensional Adaptations for Open Learning Management Systems. In: Proceedings of the 2008 IEEE/WIC/ACM Joint International Conference on Web Intelligence and Intelligent Agent Technology 2008, International Workshop TUMAS-A 2008: Web Intelligence & Intelligent Agent Technology in eLearning (in press, 2008)
22. Santos, O.C., Rodriguez Ascaso, A., Boticario, J.G., Martin, L.: User modeling for attending functional diversity for ALL in Higher Education. In: Weske, M., Hacid, M.-S., Godart, C. (eds.) WISE Workshops 2007. LNCS, vol. 4832, pp. 301–312. Springer, Heidelberg (2007)
23. Santos, O.C., Boticario, J.G., Fernandez del Viso, A., Perez de la Cámara, S., Rebate Sanchez, C.: Basic skills training to disabled and adult learners through an accessible e-Learning platform. In: 12th International Conference on Human-Computer Interaction, pp. 796–805 (2007)