
Nicola Capuano, Saverio Salerno
University of Salerno, Fisciano, Salerno, Italy
{ncapuano, salerno}@unisa.it

Giuseppina Rita Mangione, Anna Pierri
Centre of Research in Pure and Applied Mathematics
CRMPA, Fisciano, Salerno, Italy
{mangione, pierri}@crmpa.unisa.it

Abstract—The construction of personalized courses taking into account needs and preferences of each learner is one of the most studied topics within the field of adaptive learning systems. Such systems assume that learning needs are well know by the teacher or, in self directed learning settings, that learners can easily determine and express own needs. Nevertheless, in real situations, learning needs often remain latent and, for this reason, unsatisfied. To overcome this limitation we propose in this paper a methodology and a prototype system enabling the elicitation of latent learning needs, as well as the automatic generation of learning experiences capable of satisfying such needs. The proposed methodology applies principles of recommending systems and also relies on a semantic representation of topics to be taught. The encouraging experimental results obtained with University students are also discussed.

Keywords—learning needs; recommender systems; self regulated e-learning; adaptive learning systems.

I. INTRODUCTION

Adaptive learning systems have revolutionized online education by providing individualized and personalized instruction for each learner. Such systems are able to build personalized learning experiences starting from learning goals to be achieved and taking into account not only the preceding knowledge of each learner, but also the personal learning style and, in some cases, the context where the learning takes place [1].

In some systems, the course generation process can also start form an explicit request in terms of needs to be satisfied and expressed directly by the learner in natural language. This is the case of IWT (Intelligent Web Teacher), an existing adaptive learning system that we used as starting point for our research [2].

This important feature is capable of supporting the so-called self-regulated learning. This approach postulates that successful learning with modern technologies is based on the premise that learners are able to adaptively regulate their cognitive and metacognitive behaviours during the learning experience [3]. A lack of self-regulation implies deficient learning gains and conceptual understanding.

The work presented in this paper go further and tries to discover latent learning needs and automatically generate and recommend to the learner, learning experiences able to satisfy such needs. The learning need elicitation process is based on the adoption and the adaptation, to the e-learning domain, of methodologies and algorithms coming from the area of recommender systems.

In the context of self-regulated learning, such kind of educational recommendations will serve as a pedagogical advance organizer for the learners, as they anticipate and spread needs, knowledge and learning paths. Furthermore they are also capable of improving the students’ control over learning.

The paper is organized in this way: the section 2 introduces some background about recommender systems; the section 3 provides a small survey of related work; the section 4 briefly introduces IWT, the starting point of our research; the section 5 describes the defined methodology; the section 6 describe the developed prototype; the section 7 presents some experimentation result with real users; the section 8 presents conclusions and planned future work.

II. BACKGROUND

Recommender Systems (RSs) are aimed at providing suggestions on the utility of a set of objects belonging to a given domain, starting from available information about users and objects [4]. The recommendation problem can be expressed in these terms: $C$ is a set of users, $I$ is a set of objects that can be recommended, $R$ a totally ordered set whose values represent the utility of an object for a user and $u$: $C \times I \rightarrow R$ a utility function that measures how a given object $i \in I$ is useful for a user $c \in C$.

The purpose of the system is to recommend to each user $c$ the object $i$ that maximizes the utility function so that:

$$i'_c = \text{argmax}_i u(c, i).$$

The central problem is that $u$ is not completely defined on $C \times I$. In fact, in typical applications, a user never expresses preferences on each available object. A RS shall then be able to estimate the values of $u$ in the space of data where it is not defined, extrapolating it from the points of $C \times I$ where it is known.

Several RS techniques exist in literature. In cognitive approaches [5], the value of the utility function $u(c, i)$ is predicted considering the values $u(c, i_j)$ assigned to items found similar to $c$. Each object $i \in I$ is associated with a profile i.e. a vector $\text{content}(i) = (w_{i,1}, ..., w_{i,j})$ where $w_{i,j}$ is the weight of the $j$-th attribute or an indication of how the $j$-th attribute is able to characterize $i$.

As for objects, users are also associated with a profile based on the attributes of the objects preferred in the past.
The profile is defined as \( \text{profile}(c) = (w_{c,1}, \ldots, w_{c,d}) \), where each weight \( w_{c,j} \) denotes the importance of the \( j \)-th attribute for the user \( c \). The profile for \( c \) can be obtained averaging all profiles of the objects for which \( c \) has expressed a rating and weighting them on the basis of the rating itself.

Once the profiles that characterize objects and users have been defined, the utility of an object \( i \) for the user \( c \) is calculated basing on the similarity between the two profiles. Several similarity measures can be used for this purpose: one of the most common is the so-called \textit{cosine similarity} based on the calculation of the cosine between two vectors using the following equation:

\[
    u(c, i) = \frac{\sum_{j=1}^{d} w_{c,j} w_{i,j}}{\sqrt{\sum_{j=1}^{d} w_{c,j}^2} \sqrt{\sum_{j=1}^{d} w_{i,j}^2}} \tag{1}
\]

In collaborative approaches, unknown values of \( u(c, i) \) are estimated from those made available by people similar to \( c \) [6]. The basic idea is that users who evaluated in the same way the same objects are likely to have the same tastes (and are therefore similar). Such methods calculate the utility \( u(c, i) \) as aggregation of the utility expressed for \( i \) by users similar to \( c \):

\[
    u(c, i) = \text{aggr}_{C' \in C} u(c', i)
\]

where \( C' \) is the set of users considered most similar to \( c \). A typical aggregation function is the average of ratings given to \( i \) by users of \( C' \) weighted on the similarity of such users with \( c \):

\[
    u(c, i) = \frac{\sum_{c' \in C} u(c', i) \text{sim}(c, c')}{\sum_{c' \in C} \text{sim}(c, c')} \tag{2}
\]

where \( \text{sim}(c, c') \) is the degree of similarity between \( c \) and \( c' \) calculated using similarity measures like equation (1) applied to vectors \((w_{c,1}, \ldots, w_{c,d})\) that characterize users, where \( w_{c,i} = u(c, i) \), if defined.

The main advantage of collaborative approaches is that they are able to provide less obvious advice with respect to cognitive ones. On the other end cognitive approaches are able to provide useful recommendations also with only one assessment made by the user while collaborative ones need a substantial number of assessments available.

### III. RELATED WORK

Several RSs for e-Learning have been proposed till now to suggest suitable learning resources to students. One of the first collaborative recommenders for learning resources has been Altered Vista [7] whose goal was to collect user-provided evaluations about learning resources, and to use them to recommend both interesting resources and people with similar tastes and beliefs.

A similar RS is RACOFI (Rule-Applying Collaborative Filtering) [8] that combines a collaborative filtering engine, that works with ratings that users provide for learning resources, with an inference rule engine that is mining association rules between the learning resources.

CYCLADES [9] is a general recommendation service. It uses a collaborative filtering technique with user-based ratings on digital resources available in an open archive. The advantage of the system is the possibility of offering recommendations for learning activities that are developed by different institutions. This approach is currently used by the Open Education Resources movement.

A related system is CoFind [10] that uses digital resources freely available on the Web but it follows an approach that uses folksonomies for recommendations.

A different approach to RS has been followed in [11] by developing a recommender system for learning objects that is based on sequencing rules that help users be guided through the concepts of an ontology of topics. Rules are fired when gaps in the competencies of the learners are identified, and then appropriate resources are proposed.

A similar system was introduced by LSRS (Learning Sequence Recommendation System) [12]. It analyses group learning experiences to predict a personal list for each learner by tracking others’ learning patterns regarding certain topics. It proposes learning mechanism that uses Markov chains to calculate transition probabilities of possible learning objects in a sequenced course of study.

The Re-Mashed Personal Learning Environment [13] recommends learning resources from a learning network. In such system learners can specify Web 2.0 services like Flickr, delicious.com or slideshare.com and mash-up them in a personal learning environment. Learners can rate such information and train a recommender system for their particular needs.

### IV. THE STARTING POINT

The starting point of this research is an existing e-learning system named IWT (Intelligent Web Teacher). As described in [14] IWT allows the generation of personalized learning experiences basing on four models.

The \textit{domain model} describes the knowledge that is object of teaching through a set of concepts and a set of relations between concepts. Teaching preferences can be added to the domain model to define feasible teaching strategies that may be applied for available concepts.

The \textit{learner model} represents a learner through a cognitive state that measures the knowledge reached by him at a given time and by a set of learning preferences that provide an evaluation of which learning strategies are more feasible for him. Both components are automatically assessed by IWT by analysing results of testing activities and the learner behaviour during the learning experience.

The \textit{learning resource model} represents a learning resource through feasible metadata. It includes the set of concepts that are covered by the learning resource and a set of didactical properties representing learning strategies applied by the learning resource.

The \textit{unit of learning model} represents a sequence of learning resources needed for a learner in order to understand a set of target concepts in a given domain.
In [15] we have described the process able to generate a unit of learning starting from a set of a target concepts and from a learner model. The process generates a feasible sequence of domain concepts able to teach the target concepts. Then it removes domain concepts already known by the target learner by looking at his/her cognitive state. Eventually it associates to each remaining concept the best matching learning resources taking into account teaching and learning preferences.

To simplify user interactions with the system, IWT also implements an alternative method for the expression of a learning need through Upper Level Learning Goals (ULLGs) [16]. An ULLG is a meaningful set of target concepts on a given domain model with a connected textual description. ULLGs can be built either by teachers and by learners and are accessed through a search engine. The learner can so specify a learning need in natural language and let the system find the list of best matching learning resources taking into account teaching and learning preferences.

To estimate the utility of remaining concepts, an hybrid recommendation algorithm combining a cognitive component and a collaborative one is used. The cognitive component suggests concepts that are complementary to those already known or under learning. It is based on the analysis of available domain models.

Given a set of concepts \( C \) and a set of learners \( L \), the cognitive state of a learner \( l \in L \) describes the knowledge reached by \( l \) at a given time and it is represented as an application \( CS_l : C \rightarrow [0, 10] \). Given a concept \( c \), \( CS_l(c) \) indicates the degree of knowledge (or grade) reached by the learner \( l \) for \( c \). If such grade is greater then a threshold \( \theta \) then \( c \) is considered as known. The algorithm defined to calculate the cognitive state is described in [15].

At a given time a learner can be enrolled to one or more units of learning. A unit of learning represents a sequence of learning resources needed by a learner to understand a set of target concepts in a given domain. Among the components of a unit of learning there is the learning path \( LPath = (c_1, \ldots, c_n) \): an ordered sequence of concepts that must be taught to a specific learner in order to let him/her complete the unit.

Starting from that, we define the \( COT_l \) as the set of concepts that are object of teaching for a given learner as the union of all learning paths of the units of learning he is enrolled in. Then we can define the concept mapping function \( CMF : L \times C \rightarrow [0, 1] \) as follows:

\[
CMF(l, c) = \begin{cases} 
1 & \text{if } CS_l(c) \geq \theta \\
1/2 & \text{if } CS_l(c) < \theta \land c \in COT_l \\
0 & \text{otherwise}
\end{cases}
\]

So, given a learner, the \( CMF \) is 1 for all concepts that he knows and 0.5 for all concepts that are currently under learning. It is 0 for other concepts. The concept mapping function represents an implicit rating given by a learner to available concepts: concepts that are relevant for him because learnt or under learning are positively evaluated while for other concepts the evaluation is 0.

### V. THE PROPOSED APPROACH

Within this research we have defined a new process of course building based on ULLG but starting from an implicit request rather than from an explicit one. In other words, a methodology to recommend ULLGs is provided. This is based, from one side, on the analysis of a learner’ cognitive state and on the comparison of it with cognitive states of similar learners and, from the other side, on the analysis of domain models organizing concepts.

To do that we adopt a hybrid recommendation strategy that combines a collaborative approach (answering to the question: what similar users know?) with a cognitive semantic-aware one (answering to the question: which concepts complement the current knowledge according to available domain models?).

The proposed RS (that extends the work described in [17]) consists of the following steps.

- **Concept mapping**: for each learner, known concepts plus concepts currently under learning are identified.
- **Concept utility estimation**: for each learner, the utility of each unknown concept is estimated.
- **Learning goal utility estimation**: the utility of each available ULLG is calculated for each learner.

Once the process is done for a given learner, ULLGs with the greatest utility can be suggested to him. The following paragraphs deal with the description of each of these steps.

### A. Concept Mapping

Given a set of concepts \( C \) and a set of learners \( L \), the cognitive state of a learner \( l \in L \) describes the knowledge reached by \( l \) at a given time and it is represented as an application \( CS_l : C \rightarrow [0, 10] \). Given a concept \( c \), \( CS_l(c) \) indicates the degree of knowledge (or grade) reached by the learner \( l \) for \( c \). If such grade is greater then a threshold \( \theta \) then \( c \) is considered as known. The algorithm defined to calculate the cognitive state is described in [15].

At a given time a learner can be enrolled to one or more units of learning. A unit of learning represents a sequence of learning resources needed by a learner to understand a set of target concepts in a given domain. Among the components of a unit of learning there is the learning path \( LPath = (c_1, \ldots, c_n) \): an ordered sequence of concepts that must be taught to a specific learner in order to let him/her complete the unit.

Starting from that, we define the \( COT_l \) as the set of concepts that are object of teaching for a given learner as the union of all learning paths of the units of learning he is enrolled in. Then we can define the concept mapping function \( CMF : L \times C \rightarrow [0, 1] \) as follows:

\[
CMF(l, c) = \begin{cases} 
1 & \text{if } CS_l(c) \geq \theta \\
1/2 & \text{if } CS_l(c) < \theta \land c \in COT_l \\
0 & \text{otherwise}
\end{cases}
\]

So, given a learner, the \( CMF \) is 1 for all concepts that he knows and 0.5 for all concepts that are currently under learning. It is 0 for other concepts. The concept mapping function represents an implicit rating given by a learner to available concepts: concepts that are relevant for him because learnt or under learning are positively evaluated while for other concepts the evaluation is 0.

### B. Concept Utility Estimation

The utility \( u(l, c) \) of a concept \( c \) for a learner \( l \) can be estimated starting from the concept mapping function. The utility of a known concept or of a concept that will be known soon is settled to 0 because it should be excluded from any suggestion. So \( CMF(l, c) > 0 \rightarrow u(l, c) = 0 \).

To estimate the utility of remaining concepts, an hybrid recommendation algorithm combining a cognitive component and a collaborative one is used. The cognitive component suggests concepts that are complementary to those already known or under learning. It is based on the analysis of available domain models.

Given a domain model \( d \), we indicate the set of its concepts as \( C_d \). We then define the cognitive utility of a concept \( c \) belonging to \( C_d \) by the following way:

\[
u_{cog}(l, c) = \frac{CMF(l, c)}{|C_d|}.
\]
The value of $u_{cog}(l, c)$ does not depend on $c$ itself but on the domain $d$ it belongs to. It is proportional to the number of concepts $l$ has acquired in $d$ normalized on the total number of concepts belonging to the same domain.

The **collaborative component** of the utility of a given concept $c$ for a learner $l$ is then estimated by aggregating ratings for the concept $c$, included in the concept mapping function, coming for learners that are similar to $l$. The estimation is done by adapting the equation (2):

$$u_{cog}(l, c) = \frac{\sum_{c' \in L'} CMF(l', c) \cdot \text{sim}(l, l')} {\sum_{c' \in L'} \text{sim}(l, l')}$$

where $L'$ is the set of the learners most similar to $l$ while $\text{sim}(l, l')$ is the similarity degree between the learner $l$ and the learner $l'$ obtained though similarity measures described in section 2.

After having calculated cognitive and collaborative components, the utility of a concept $c$ for a learner $l$ is estimated by hybridizing these values and by letting it equal to 0 for concepts already known or that will be known soon according to the concept mapping function:

$$u(l, c) = \begin{cases} 
\alpha u_{cog}(l, c) + (1 - \alpha) u_{coll}(l, c) & \text{if } \text{CMF}(l, c) = 0 \\
0 & \text{otherwise}
\end{cases}$$

where $\alpha$ is the hybridisation coefficient that is a real number between 0 (highest priority to the collaborative component) to 1 (highest priority to the cognitive one). Low values for $\alpha$ privileges serendipity while high values privilege suggestions’ accuracy.

**C. Learning Goal Utility Estimation**

In [14] it is explained how to generate a learning path starting from a set of target concepts. By applying the algorithm described there, it is possible to determine, for each $ULLG_i$, the corresponding learning path $LPath_i$. Then it is possible to estimate the **conceptual utility** of each of them for a learner $l$ with the following equation:

$$u_{con}(l, ULLG_i) = \sum_{c \in LPath_i} u(l, c).$$

The calculation of the conceptual utility takes into account the utility of all concepts explained by the ULLG, including those already known by the learner, if any. To take into account only unknown concepts, we calculate the **marginal utility** obtained in this way:

$$u_{mar}(l, ULLG_i) = \sum_{c \in LPath_i \setminus \{c' \in LPath_i \mid \text{CMF}(l, c') = 0\}} u(l, c).$$

Thus the utility of an ULLG for a given learner is obtained by combining conceptual and marginal utilities through a weighted sum with the following equation:

$$u(l, ULLG_i) = \beta u_{con}(l, ULLG_i) + (1 - \beta) u_{mar}(l, ULLG_i).$$

where $\beta$ is the hybridisation coefficient: a real number between 0 (highest priority to the marginal utility) to 1 (highest priority to the aggregated utility). Low values for $\beta$ privileges accuracy while high values privilege novelty.

A baseline recommender is activated when the system has no suggestions for a given user i.e. when the utility of every ULLG for him is zero (this usually happen for new learners). In such case the average social rating associated to each ULLG is used as an estimation of the popularity i.e. the average perceived utility of this ULLG:

$$u(l, ULLG_i) = \frac{r(ULLG_i)}{10}$$

where $r(ULLG_i)$ is the average social rating of $ULLG_i$, ranging from 0 to 5 (5-star scale).

**VI. The Prototype**

In order to experiment the proposed approach, we designed, developed and integrated with the IWT system (in particular with the Learning Goals Manager module), two prototype components for learning goals indexing and recommendation as shown in figure 1 (where black boxes represent pre-existing IWT components while grey ones are new components introduced by us).

The **learning goals indexer** works in background to maintain the concept mapping matrix and the user-to-user similarity matrix. The **learning goals recommender** works in real time to calculate the utility of each concept and of each available ULLG basing on data maintained by the indexer and exploiting cognitive and collaborative sub-components in order to suggest the most feasible ULLGs for each learner.

---

**Figure 1. The prototype architecture.**
The Figure 2 shows the developed Personal Learning Needs panel dealing with ULLG management. The first section is titled My Learning Goals. Here a learner can view and manage his/her ULLGs. By pressing the access link he can follow the personalized unit of learning connected with each learning goal. By pressing the remove link he can unsubscribe the course connected with the ULLG. He can also rate the learning goal through a standard 5-star scale.

When icons representing subscribed learning goals exceed the space available in the panel section a more link appears in the lower-right corner of the section. Once this link is followed, a new page appears and all subscribed ULLGs are displayed. Searching and filtering facilities are provided here.

The second section is titled Recommended Learning Goals. Here a learner can view the list of learning goals that are suggested by the system for him by applying the defined methodology. By pressing the details link he can obtain more details about the ULLG. By pressing the add link, the learning goal is added to the My Learning Goals section and the connected unit of learning is subscribed.

Under each ULLG, the learner can view the average rating provided by other people that have already used it. Additional suggestions can be accessed by clicking on the more link.

In the third section titled Express a New Learning Need the learner can write a need in natural language and let the system find a list of suitable ULLGs by pressing the button Search an Existing Learning Goal. The learner can also build a new learning goal based on the written text by pressing the button Build a New Learning Goal.

In the latter case a list of concepts related to the query taken from available domain models is presented to the learner that can add or remove some of them to a new ULLG. Then the learner can add a description and save the new ULLG that is added to the My Learning Goals section with a different icon. The created ULLG can be shared with other learners.

VII. EXPERIMENTATION RESULTS

To evaluate the implemented prototype as well as the underlying methodology and to analyse its effects in the learning process we experimented it with real users within an University setting. In particular 61 students enrolled in an on-line course on Software Engineering were involved in the experiment, 29 of them in the experimental group and 32 in the control group.

Students of the experimental group were enabled to the Personal Learning Need prototypical section of the learning system where to find tailored recommendation about how to complement their current knowledge with additional topics related to Software Engineering. The same section was inhibited to control group students.

After the assignment, students of the experimental group were required to fill a questionnaire that included the following sections: (1) test-based evaluation of system usability; (2) test-based evaluation of the validity of the proposed approach; (3) evaluation questions about the knowledge acquired with the course. Students belonging to the control group had to fill only the third section.

To evaluate system usability we used the System Usability Scale (SUS) [18] which contains 10 items and a 5 point Likert scale to state the level of agreement or disagreement. SUS scores have a range of 0 to 100 with an average score of 68, obtained from 500 studies. After calculating the SUS score for each student, we got an average of 53.97 that is quite encouraging considering the prototypical nature of the system.

Analyzing student feedback, we can observe there are many students who think they would like to use the system more often (M=3.13, SD=1.09, Md=3). A lot of students thought that the system is easy to use (M=3.65, SD=0.81, Md=4). In addition, many students stated that they had not needed the support of a technician to be able to use the system and that people should learn how to use it quickly (M=2.90, SD=1.01, Md=3).

To evaluate the validity of the approach we asked students to answer 3 qualitative high-level questions with a scale ranging from 0 to 10. By analysing obtained results we can observe that many students have found the system useful for their study (M=6.21, SD=2.02, Md=6). Most of them also agreed that recommended learning goals are useful and correctly complement topics studied within the course (M=6.17, SD=2.37, Md=6).

A lot of students also considered the self-evaluation questionnaires included within recommended units of learning as being very important for their learning process in order to clarify doubts and to assimilate concepts. (M=6.21, SD=1.97, Md=6).

All students from both the experimental and control groups were evaluated basing on the answers provided to a questionnaire purposed to evaluate acquired knowledge. To this end 2 questions about software engineering topics have been included, the first more general and practical, the second more specific and theoretical.
This part of each questionnaire was assessed by a lecturer who used the standard 10-point scale to score the students’ responses. Table 1 shows the results. Obtained results show that students from the experimental group scored slightly higher than the control group.

### VIII. CONCLUSIONS AND FUTURE WORKS

We defined in this paper a methodology to recommend learning goals basing on similarities between users as well as on semantic structures representing teaching domains. A prototype component applying the defined methodology was developed and integrated within an existing e-learning system in order to experiment it with real users.

Experimentation results have shown that in general the students liked the tool and found interesting and useful to have personalized recommendations complementing their knowledge about topics under study. In particular obtained units of learning fulfilled the expectations of the learners. In addition, in line with the prototypical nature of the system, the usability was not a barrier when using it.

The validation of the effectiveness of the proposed system in learning of scientific concepts was also analysed and was observed that grades obtained by students of the experimental group were slightly higher with respect to those belonging to the control group.

Future work will include a study purposed to improve the usability of developed prototype. From the other side, the application of matrix factorisation techniques [19] will be considered to optimize recommender performances.

## ACKNOWLEDGMENT

This research is partially supported by the European Commission under the Collaborative Project ALICE [20] “Adaptive Learning via an Intuitive, interactive, collaborative, Emotional system”, 7th Framework Program, Theme ICT-2009.4.2, Grant Agreement n. 257639.

## REFERENCES


