Computers in Human Behavior 30 (2014) 663–673

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Elicitation of latent learning needs through learning goals recommendation

Nicola Capuano [a], Matteo Gaeta [b], Pierluigi Ritrovato [b,*], Saverio Salerno [b]

[a] Centro di Ricerca in Matematica Pura ed Applicata, Fisciano SA, Italy
[b] University of Salerno, Fisciano SA, Italy

ARTICLE INFO

Article history:
Available online 1 October 2013

Keywords:
Adaptive learning
Recommender systems
Intelligent tutoring systems

ABSTRACT

The aim of a recommender system is to estimate the relevance of a set of objects belonging to a given domain, starting from the information available about users and objects. Adaptive e-learning systems are able to automatically generate personalized learning experiences starting from a learner profile and a set of target learning goals. Starting form research results of these fields we defined a methodology and developed a software prototype able to recommend learning goals and to generate learning experiences for learners using an adaptive e-learning system. The prototype has been integrated within IWT: an existing commercial solution for personalized e-learning and experimented in a graduate computer science course.

1. Introduction

A significant educational action able to guide the learner in a comprehensive learning process is not only focused on learning (cognition level) but also on fostering a correct learning behavior that empowers learners to achieve their learning goals in a controlled and directed way (metacognition level) (Mangione, Gaeta, Orciuoli, & Salerno, 2010).

Starting from this principle we defined and developed an e-learning system able to build personalized learning experiences starting from a set of target concepts selected on an ontology-based domain model (Capuano, Gaeta, Miranda, Orciuoli, & Ritrovato, 2008). We then extended such system in order to allow course generation from an explicit request in terms of needs to be satisfied and expressed by the learner in natural language (CapuaCapuano, Gaeta, Orciuoli, & Ritrovato, 2009).

The work presented in this paper deals with the definition of a further process of course building starting from an implicit request rather than from an explicit one. In other words, a methodology to recommend learning goals based on the analysis of a learner' profile (including known topics) and on the comparison of this profile with profiles of similar learners is defined.

The proposed methodology upholds the social presence (Acampora, Gaeta, Orciuoli, & Ritrovato, 2010; Capuano, Gaeta, Orciuoli, & Ritrovato, 2010), while supporting the development of self-regulated learning. Educational recommendations serves as a pedagogical advance organizer for the learners, as it anticipates and spreads needs, knowledge and learning paths. Furthermore the proposed solution also supports help seeking processes improving the students' control over learning. This makes the solution adequate nononly for educational settings but also for enterprise training (Capuano, Gaeta, Ritrovato, & Salerno, 2008).

The paper is organized in this way: Section 2 introduces some background about recommender systems; Section 3 briefly introduces the starting point of our research and then describes the proposed methodology; Section 4 introduces the developed prototype and presents some example of use; Section 5 compares our approach with some existing recommender systems for e-learning; eventually Section 6 presents conclusions and planned future work.

2. Background on recommender systems

Recommender Systems (RS) are aimed at providing personalized recommendations on the utility of a set of objects belonging to a given domain, starting from the information available about users and objects.

A formal definition of the recommendation problem can be expressed in these terms (Adomavicius & Tuzhilin, 2005): C is the set of users of the system, I the set of objects that can be recommended, R a totally ordered set whose values represent the utility of an object for a user (e.g. integers between 1 and 5 or real numbers between 0 and 1) and u: $C \times I \rightarrow R$ a utility function that measures how a given object $i \in I$ is useful for a particular user $c \in C$. The purpose of the system is to recommend to each user c the object i'_c that maximizes the utility function so that:

* Correspoding author.
E-mail addresses: capuano@crmpa.unisa.it (N. Capuano), mgaeta@unisa.it (M. Gaeta), pritrovato@unisa.it (P. Ritrovato), salerno@unisa.it (S. Salerno).

0747-5632/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.chb.2013.07.036

$$i'_c = \arg\max_{i \in I} u(c, i) \quad (1)$$

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

In other words, the goal is to make a prediction about the vote that a particular user would give to an object that has not been rated yet. Several techniques for RS exist in literature, they are usually divided in three broad categories:

- *cognitive (or content-based) approaches:* specific objects are recommended to the user, similar to those that have been positively rated in the past (they are therefore based on the calculation of similarity between objects);
- *collaborative approaches:* specific objects are recommended to the user, in particular those objects that are liked by other people with similar tastes (they are therefore based on the calculation of similarity between users);
- *hybrid systems*: they combine the two previous approaches.

In following sub-sections, we present the three approaches by considering the advantages and disadvantages of each of them. As described in Section 3, our methodology applies a hybrid approach, combining cognitive and collaborative elements.

2.1. Cognitive approaches

In cognitive approaches (Balabanovic & Shoham, 1997), the value of the utility function $u(c, i)$ of the user c for the object i is predicted by considering the values $u(c, i_k)$ to be assigned to items found similar to c. In general, each object $i \in I$ is associated with a profile, i.e. a set of attributes able to characterize the content, that is represented by a vector $content(i) = (w_{i,1}, \ldots w_{i,k})$ 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 the object i. The attributes' weight can be created automatically by the system or manually by a user.

As for the objects, users are also associated with a profile based on the attributes of the objects preferred in the past. The profile is defined as $profile(c) = (w_{c,1}, \ldots w_{c,k})$, where each weight $w_{c,j}$ denotes the importance of the j-th attribute for the user c. The profile of user c can be obtained, in the simplest formulation, averaging all profiles of the objects for which c has expressed a rating and weighting them on the basis of the rating itself. Obviously, the profile varies over the time depending on the assessments that the user gradually provides.

Once the profiles that characterize objects and users have been defined, the utility of an object i for user c is calculated basing on the similarity between the two profiles. In other words $u(c, i) = sim(profile(c), content(i))$. Several similarity measures can be used for this purpose: one of the most common is the so-called cosine similarity based on the calculation of the cosine between two vectors using the following formula:

$$Sim(profile(c), content(i)) = \frac{\sum_{j=1}^{k} W_{c,j} W_{i,j}}{\sqrt{\sum_{j=1}^{k} W_{c,j}^2} \sqrt{\sum_{j=1}^{k} W_{i,j}^2}} \quad (2)$$

The main advantage of cognitive approaches is that the recommendations are only based on data related to the domain objects: first useful recommendations are then made immediately, with only one assessment made by the user. This feature is important in environments where it is necessary to produce immediate results or in which new users are added frequently. On the other hand this approach tends to over-specialize predictions, therefore making them uninteresting.

2.2. Collaborative approaches

In collaborative approaches, unknown values of the utility function $u(c, i)$ are estimated from those made available by people considered similar to c (Konstan et al., 1997). 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).

Collaborative systems are very popular and are classified in categories depending on the algorithm used to explore the connections between users. Among the others, user-to-user memory-based algorithms (Perugini, Goncalves, & Fox, 2004) calculate the utility $u(c, i)$ as aggregation of the utility expressed for i by users similar to c; in other words:

$$u(c, i) = aggr_{c' \in C'} u(c', i) \quad (3)$$

where C' is the set of n users considered most similar to c (with n chosen between 1 and the total number of system users).

The simplest aggregation function is the average of ratings given to the users of C' or, as expressed below, the average of such ratings weighted on the degree of similarity between users who have expressed them:

$$u(c, i) = \frac{\sum_{c' \in C'} u(c', i) \cdot sim(c, c')}{\sum_{C' \in C'} |sim(c, c')|} \quad (4)$$

where $sim(c, c')$ indicates the degree of similarity between users c and c' calculated using similarity measures such as the cosine similarity (2) or the Pearson's correlation coefficient (Adomavicius & Tuzhilin, 2005). These measures are applied to vectors $(w_{c,1}, \ldots, w_{c,m})$ that characterize users, where $w_{c,i} = u(c,i)$, if defined.

By computing recommendations basing on the similarity between users, the advantage is to provide more accurate and less obvious advice. Conversely, the main problem occurs in domains with a large number of objects and/or users. Preferences in such environments are extremely sparse and the utility function is defined on a tiny part of the space $C \times I$. In these scenarios, it is difficult to calculate the correlation between users, so the recommendations are generated in an inaccurate way.

Directly linked to this limit, there is the commonly called *cold start problem*, that occurs in the early days of life of a system, when the available number of assessments is still lower than those of a fully operational system.

2.3. Hybrid approaches

Hybrid approaches try to overcome problems of both cognitive and collaborative approaches by using the two techniques simultaneously. There are several methods by which collaborative and cognitive approaches may be combined into a single system. Among them we quote the following (Burke, 2007):

- *weighted hybridization* (a cognitive and a collaborative algorithms are developed and, as final result, a combination of predictions from the two is used);
- *switching* (it is like the previous one but the system chooses, as appropriate, only one algorithm among those developed and it only returns results from it);
- *cascade hybridization* (available algorithms are ranked in order of priority and lower-level ones can only refine the results calculated from higher-level ones);
- *ad hoc algorithms* (they are specific implementations that combine cognitive and collaborative elements).

In general, hybrid recommender systems have, at the same time, the benefits of cognitive and collaborative systems. The downside is, of course, that these benefits are mitigated as a result of the composition.

3. The proposed approach

In this section we describe the methodology we have defined to recommend learning goals to users of an existing learning system named IWT (Intelligent Web Teacher). First of all a brief introduction to IWT is provided in the next sub-section as well as some fundamentals on *Upper Level Learning Goals* (ULLGs): a user friendly way (using natural language) for the expression of learning needs provided by IWT.

After having introduced the starting point, a methodology to recommend ULLGs basing on the analysis and the comparison of learners' knowledge is provided. The algorithm applies an hybrid approach to recommendation consisting of three steps each described in a separate sub-section. It improves preliminary results obtained so far and discussed in Capuano, Mangione, Pierri, and Salerno (2012).

3.1. The starting point: IWT

In this section we introduce the learning system named IWT (Albano, Gaeta, & Ritrovato, 2007) that we adopted as a basis to apply models and methodologies hereafter defined. As described in Capuano et al. (2008), IWT allows to generate personalized learning experiences and relies on four interacting models as described below. In (Gaeta, Orciuoli, & Ritrovato, 2009) it is also presented the integrated approach to manage the life-cycle of ontologies, used to define personalized e-Learning experiences.

The *domain model* describes the knowledge that is object of teaching through a set of concepts (representing topics to be taught) and a set of relations between concepts. A set of teaching preferences can be added to the domain model to define feasible teaching strategies that may be applied for each available concept.

The *learner model* represents a learner and is composed by 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 analyzing results of testing activities and the learner behavior during the learning experience.

The *learning resource model* is a metadata describing a learning resource using the IEEE LOM standard (IMS Global Learning Consortium., 2006). It includes the set of concepts that are covered by the learning resource and an additional set of didactical properties representing learning strategies applied by the learning resource.

The *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 (Capuano, Gaeta, Salerno, & Mangione, 2011) we have described the process to generate a personalized and contextualized unit of learning starting from a set of target concepts and from a learner model. The process generates a feasible sequence of domain concepts able to learn the target concepts. Then it removes domain concepts already known by the 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) Gaeta, Orciuoli, Paolozzi, & Ritrovato, 2009. 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 or 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 ULLGs basing on the similarity between the expressed need and the textual descriptions connected to ULLGs. Then the learner can select a ULLG and let the system build a personalized unit of learning starting from the connected set of target concepts and from his/her learner model.

The next sub-sections deal with the integration in IWT of a new process for 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 based on the analysis of a learner' cognitive state and on the comparison of this cognitive state with cognitive states of similar learners is provided.

The algorithm consists of the following steps: concept mapping, concept utility estimation and ULLG utility estimation each described in one of the following sub-sections. Once the utility of each ULLG is estimated for a learner, the ULLGs with the greater utility can be suggested to him.

3.2. Phase 1: concept mapping

Given a set of concepts C and a set of learners L, the *cognitive state* of a learner $l \in L$ (as reported in 3.1), 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, with $CS_l(c)$ we indicate the degree of knowledge (or grade) reached by the learner l for c. If such grade is greater than a threshold θ then c is considered as known by l, otherwise it is considered as unknown.

At a given time a learner can be enrolled to one or more units of learning. As reported in 3.1 (and detailed in Capuano et al. (2008)), a unit of learning represents a sequence of learning resources needed by a learner in order 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 of learning.

Starting from that, we can define the set COT_l of all concepts that are object of teaching for a given learner as the union of all learning paths *LPath* corresponding to the units of learning the learner is enrolled in. Then we can define the concept mapping function (CMF) that is a Boolean function CMF: $L \times C \rightarrow \{0,1\}$ that can be defined as follows:

$$CMF(l,c) = \begin{cases} 1 \text{ if } CS_l(c) > \theta \text{ or } c \in COT_l \\ 0 \text{ otherwise} \end{cases} \quad (5)$$

So, given a leaner l, $CMF(l, c) = 1$ for all concepts c that are already known by l plus all concepts c that are currently object of teaching for him/her. It is equal to 0 for any other concepts.

3.3. Phase 2: 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 equal to 0. So $CMF(l, c) = 1 \rightarrow u(l, c) = 0$. Conversely, to estimate the utility of remaining concepts, a user-to-user recommendation algorithm is used.

We can estimate the unknown utility of a given concept c for a learner l by aggregating, through a weighted sum, ratings for the concept c, included in the concept mapping function, coming for

learners that are similar to l. The estimation can be done through the following formula, obtained as an adaptation of (4):

$$u(l,c) = \frac{\sum_{l' \in L'} CMF(l',c) \cdot sim(l,l')}{\sum_{l' \in L'} |sim(l,l')|} \quad (6)$$

where L' is the set of the n learners most similar to l while $sim(l, l')$ is the similarity degree between l and l' obtained though similarity measures like the cosine similarity reported by (2) or the Pearson correlation coefficient calculated on CMF.

From the algorithmic point of view, to estimate the concept utility function, we start from the concept mapping matrix where each element CMF(l,c) is defined with (5). This matrix is built the first time by considering every cognitive state and every course available on the system. Each time a learner starts, terminates or abandons a course then the row corresponding to this learner is updated, again, through (2).

Starting from the concept mapping matrix, the user-to-user similarity matrix is calculated. Each element $sim(l, l')$ of this matrix is obtained through a similarity measure between the rows of the concept mapping matrix corresponding to users l and l'. Once the similarity matrix is calculated, to estimate an undefined $u(l, c)$ for a given learner l, it is necessary to isolate and combine, by applying (6), the utility expressed for c by the n learners more similar to l.

3.4. Phase 3: ULLG utility estimation

An ULLG can be formally defined as a tuple $ULLG_i = (D_i, TC_{i,1}, \ldots, TC_{i,n})$ where D_i is a text describing the learning objective in natural language, while $TC_1, \ldots, TC_n$ is the list of target concepts that have to be mastered by a learner in order to reach such learning objective. A learning need LN is a textual sentence (like "to learn Java programming" or "how to repair a bicycle" etc.) expressed by a learner in order to start the unit of learning building process.

Through the unit of learning generation algorithm introduced in 3.1 (and detailed in Capuano et al. (2008)) IWT is able 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 existing upper level learning goal $ULLG_i$, the corresponding learning path $LPath_i$ starting from the connected list of target concepts.

Once determined learning paths associated to available ULLGs, it is possible to estimate the aggregated utility $au(l, ULLG_i)$ of each of them for a learner l with the following formula:

$$au(l, ULLG_i) = \sum_{c \in LPath_i} \frac{u(l,c)}{|LPath_i|} \quad (7)$$

The calculus of the aggregated utility takes into account the utility of all concepts explained by the ULLG. This means that, if the learning path connected with the ULLG includes many concepts already known by the learner, its aggregate utility can be low even if the utility of remaining concepts is high. To take into account this information we introduce the concept of **marginal utility** $mu(l, ULLG_i)$ of $ULLG_i$ for a learner l that can be obtained with the following formula:

$$mu(l, ULLG_i) = \frac{\sum_{c \in LPath_i} u(l,c) \cdot (1 - CMF(l,c))}{\sum_{c \in LPath_i} (1 - CMF(l,c))} \quad (8)$$

Thus the utility of an ULLG for a given learner can be obtained by combining aggregated and marginal utilities through a weighted sum with the following formula:

$$U(l, ULLG_i) = \alpha \, au(l, ULLG_i) + (1 - \alpha) mu(l, ULLG_i) \quad (9)$$

where α is the hybridization coefficient that is a real number between 0 (highest priority to the marginal utility) to 1 (highest priority to the aggregated utility). The choice for α is done empirically basing on experimentation results. Low values for α privileges novelty while high values privilege accuracy of suggestions given by the recommender system.

3.5. Algorithmic view

In order to further explain the recommendation building process, an algorithmic view of the whole approach is provided below. Each time a learner l starts, terminates or abandons a course the following pseudo-code is executed:

```
calculate COT_l as described in phase 1
for each concept c
   calculate CMF(l, c) by using (5)
for each learner l' different from l
   calculate sim (l, l') = sim (l', l) by using (2) on CMF
```

Each time a learner l requests a recommendation, the following pseudo-code is executed:

```
for each concept c
   calculate u(l, c) by using (6)
for each learning goal ULLG
   calculate au(l, ULLG) by using (7)
   calculate mu(l, ULLG) by using (8)
   calculate u(l, ULLG) by using (9)
```

When the utility of every available learning goal $ULLG$ for the learner l is calculated, the n ULLGs with higher values for $u(l, ULLG)$ are proposed to him. To speed-up the process, it is possible to execute the second pseudo-code for all learners as a background job on a timed schedule.

4. The developed prototype

In order to experiment the proposed approach, we designed and developed a prototype recommender system for ULLG and integrated it with IWT. In the following sub-sections we present a high-level view of the prototype architecture, give some details about its user interface and show an example of use.

4.1. The architecture

The prototype was designed and implemented as a plug-in of the IWT system. IWT architecture is divided in four main layers:

- the first layer is the *framework* used by developers to design and implement core and services, application services and learning applications;
- the second layer is composed by *core services* providing basic features to manage users, roles, resources and ontologies as well as for user profiling and learning personalization;
- the third layer is composed by *application services* used as building blocks to compose e-learning applications for specific domains including learning and content management, ontology management, ULLG management, communication and collaboration;
- on the top of the stack, *applications* covering specific scenarios obtained as integration of application services are built.

IWT server-side components are developed in Microsoft .NET technology and use Microsoft SQL Server for persistency. IWT is an extensible system both at the level of learning resources (with *drivers* i.e. software components able to edit, manage and deliver

a specific kind of resource) and at services level (with *plug-ins* i.e. software components providing specific back-end services).

IWT comes with a ULLG manager within application services. This is made of two modules: a *designer* aimed at defining an ULLG as an aggregation of a set of target concepts with a textual description and a *selector* aimed at finding the best ULLG starting from a query in a natural language.

To implement the approach described in this paper, the following two additional modules have been implemented and integrated to the ULLG manager:

- the *indexer* that works in background to maintain the concept mapping matrix and the user-to-user similarity matrix;
- the *recommender* that calculates in real time the utility of each concept and of each available ULLG basing on data structures maintained by the *indexer* and suggests the most feasible ULLGs for each learner.

The Fig. 1 shows the existing modules of the ULLG manager (in grey) and the modules that we have developed (in black). Both additional modules have been developed in C# starting from *MyMediaLite*:[1] a lightweight, multi-purpose library of recommender system algorithms.

4.2. The user interface

After having accessed the IWT system, a learner can search and use ULLGs created by different teachers going in the *formative needs* section (see Fig. 2).

Here the learner can select one of the following available options:

- *Express learning needs*: allows the learner to indicate in natural language the learning needs he wants to achieve and to verify what are the most suitable ULLGs available in the system to fulfil such needs.
- *Browse existing ULLGs*: allows the learner to view the complete collection of ULLGs created by teachers and to select the most suitable for him.
- *View recommended learning goals*: allows the learner to view a set of ULLGs the system suggests for him thanks to the developed recommender system.
- *My learning goals*: allows the learner to view and manage selected ULLGs and to study connected courses.

Clicking on the *view recommended learning goals* link, a list of recommended ULLGs is displayed to the learner, along with their relevance score by applying the methodology described in this paper. The relevance score is obtained by normalizing the ULLG utility defined by (9). Displayed ULLGs are ordered with respect to the relevance score.

The Fig. 3 shows a sample list of suggested ULLGs in the domain of computational logics. By clicking on the ULLG icon it is possible to view summary information about it. By clicking on the *delivery* link it is possible to use the ULLG i.e. the ULLG is transferred in the *my learning goals* area where the connected course can be accessed by the learner.

4.3. The prototype at work

To analyze the system behavior, we have created a sample domain for *computational logics* composed of the following concepts: *outline of set theory*, *logics*, *formal systems*, *propositional logics*, *first order logics*, *description logics* and *ontologies*. Defined concepts have been connected with appropriate relations (not reported here for brevity) according to the IWT domain model described in 3.1.

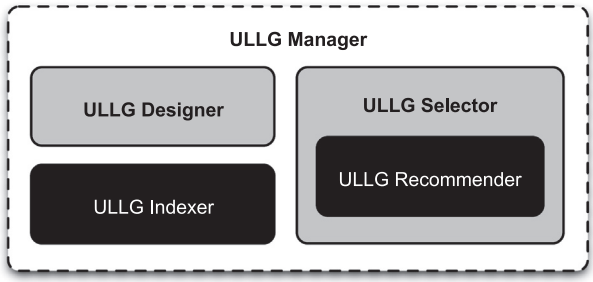

Fig. 1. Existing (in grey) and additional (in black) modules of the ULLG Manager.

Starting from defined concepts we have also created five ULLGs, targeting the following concepts: *logics*, *propositional logics*, *first order logics*, *description logics* and *ontologies*. Table 1 reports the learning path connected with each of these ULLGs obtained through the application of the learning generation algorithm.

In the system we have also created three different learners each with his own learner model. The cognitive states of such learners are summarized in Table 2. As it can be seen, the cognitive state of *learner 1* is very similar to that of *learner 2* that also knows the concept *description logic*, so it is expected that the system suggests the *ULLG 3* (about *description logics*) to *learner 1* with an higher relevance score.

But *learner 1* is also similar to *learner 3* (with a lower degree of similarity because they have less learnt concepts in common), so it is expected that the system also suggests the *ULLG 5* (about *ontologies*) to *learner 1* even if with a lower relevance score. Suggestions for *learner 1* are reported in Fig. 3. As it can be seen our expectations have been respected.

It is important to note that the system also suggest to *learner 1* the *ULLG 4* about *logics*. This is because, like *ULLG 3*, it is capable of enhancing the cognitive state of *learner 1* with the missing concept of *description logics*. Despite that the final result of the two ULLG is the same, the system gives an higher relevance score to *ULLG 3* because, even if the marginal utility of both ULLGs is the same (they add the same number of useful concepts to the cognitive state of the learner), the aggregated utility of *ULLG 3* is higher (it includes the minimum number of unneeded concepts).

Suggestions provided by the system to *learner 2* are reported in Fig. 4. It is equally similar to *learner 1* and *learner 3* so, in this case, the system suggests to him the *ULLG 5* about *ontologies* (coming from *learner 3*) and the *ULLG 2* about *first order logics* (coming from *learner 1*). *ULLG 5* has an higher relevance score with respect to *ULLG 2* thanks to an higher aggregated utility due to the inclusion of less unneeded concepts. Also *ULLG 4* about *logics* is suggested because it also covers the missing *first order logics* concept but with an even lower aggregated utility.

5. Related work

Several recommender systems for e-Learning have been introduced to select and propose learning resources to users. Some of them are still at prototype stage while some of them are full systems (Bodea, Dascalu, & Lytras, 2012). One of the first collaborative recommenders for learning resources has been Altered Vista (Recker & Wiley, 2001) whose goal was to explore how to collect user-provided evaluations about learning resources, and to use them to recommend, to the members of a community, both interesting resources and people with similar tastes and beliefs.

[1] http://www.ismll.uni-hildesheim.de/mymedialite/.

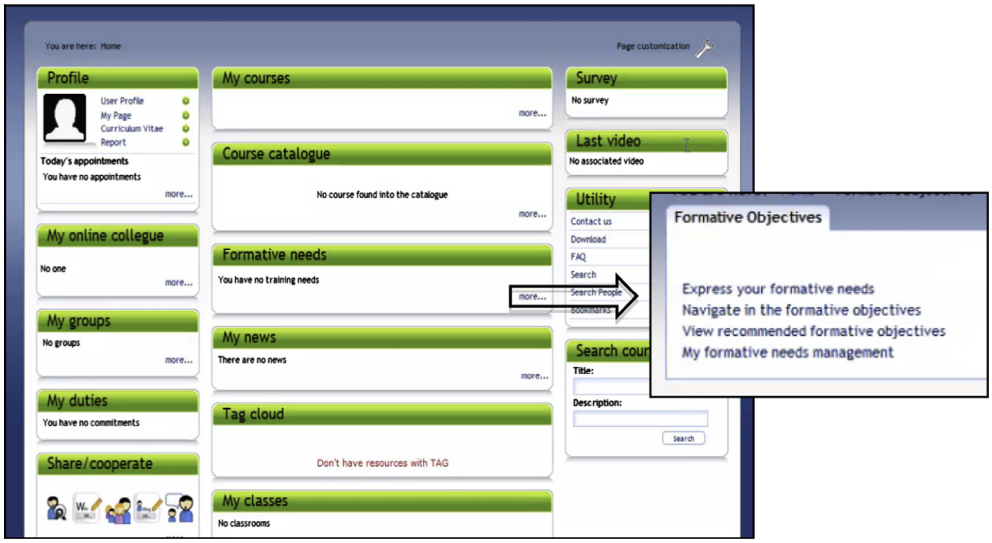

Fig. 2. The IWT user interface and the formative need section.

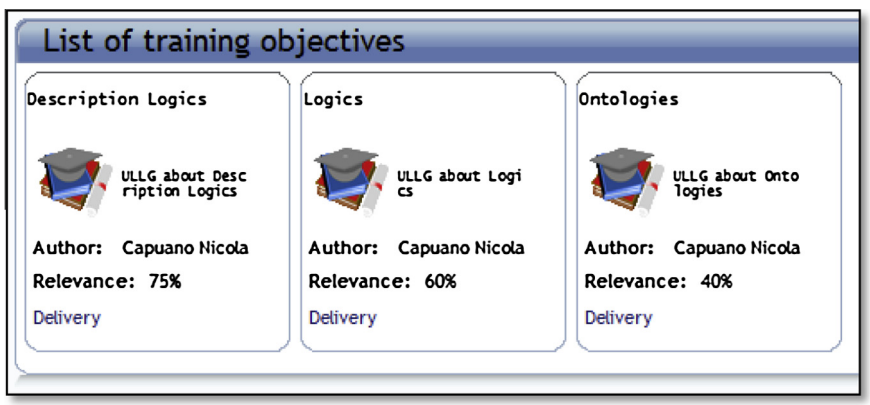

Fig. 3. A list of ULLGs suggested to a given user.

Another system that has been proposed for the recommendation of learning resources is RACOFI (Rule-Applying Collaborative Filtering) (Anderson et al., 2003) 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.

The QSIA (Questions Sharing and Interactive Assignments) system (Rafaeli, Barak, Dan-Gur, & Toch, 2004) is purposed at resources sharing, assessing and recommendation. This system is used in the context of online communities to harness the social perspective in learning and to promote collaboration, online recommendation, and learner communities building. The system has been used in several learning situations, such as knowledge sharing among faculties and teaching assistants, high school teachers and among students.

The CYCLADES system (Avancini & Straccia, 2005) is a general recommendation service. It uses a collaborative filtering technique with user-based ratings, but does not just apply the technique to one community. It uses digital resources, which are freely available in the repositories of the Open Archives Initiative.[2] 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 the CoFind prototype (Dron, Mitchell, Boyne, & Siviter, 2000). It uses digital resources that are freely available on the Web but it follows a new approach that uses folksonomies for recommendations. The CoFind developers stated that predictions according to preferences were inadequate in a learning context and therefore more user driven bottom-up categories like folksonomies are important.

A different approach to learning resources' recommendation has been followed by Shen in Shen and Shen (2004) where a recommender system for learning objects that is based on sequencing rules that help users be guided through the concepts of ontology of topics has been developed. Rules are fired when gaps in the competencies of the learners are identified, and then appropriate resources are proposed to the learners.

A similar sequencing system has been introduced in Huang, Huang, Wang, and Hwang (2009). The proposed system, the Learning Sequence Recommendation System (LSRS), analyses group-learning experiences to predict and provide 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.

[2] http://www.openarchives.org/.

Table 1
Correspondence between sample ULLGs and domain concepts.

ULLGs	Concepts					
	Outline of set theory	Formal systems	Propositional logics	First order logics	Description logics	Ontologies
ULLG 1: Propositional logics	X	X	X			
ULLG 2: First order logics	X	X		X		
ULLG 3: Description logics	X				X	
ULLG 4: Logics	X	X	X	X	X	
ULLG 5:Ontologies						X

Table 2
Cognitive states of three sample learners.

Learners	Concepts					
	Outline of set theory	Formal systems	Propositional logics	First order logics	Description logics	Ontologies
Learner 1	X	X	X	X		
Learner 2	X	X	X		X	
Learner 3	X	X			X	X

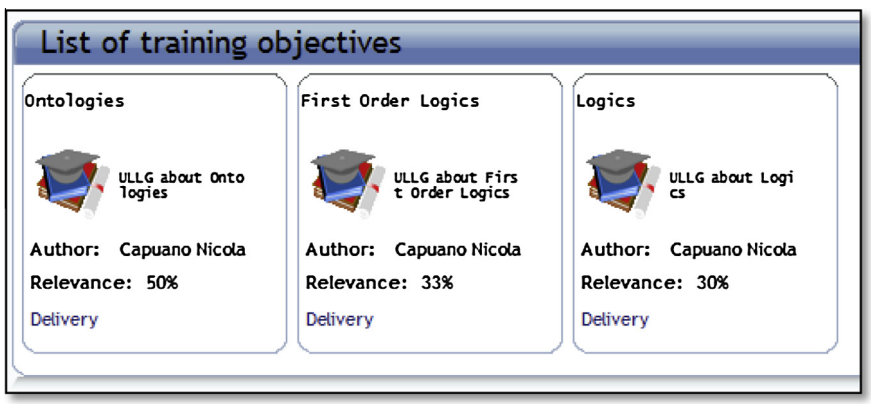

Fig. 4. The list of ULLGs suggested to the learner 2.

The Evolving e-Learning System (ELS) Tang, & McCalla, 2003 is another system including a hybrid recommendation service. The system is used for storing and sharing research papers and glossary terms among university students and industry practitioners. Resources are described (tagged) according to their content and technical aspects, but learners also provide feedback on them in the form of ratings. Recommendation takes place both using data clustering techniques to group learners with similar interests and collaborative filtering techniques to identify similar learners in each cluster.

The Re-Mashed Personal Learning Environment (Drachsler et al., 2009) recommends learning resources from emerging information of a learning network. In such system, learners can specify Web 2.0 services like *Flickr*, *delicious.com* or *sildeshare.com* and mash-up them in a personal learning environment. Learners can rate such information and train a recommender system for their particular needs.

A hybrid recommendation approach has been adopted in CourseRank (Bercovitz et al., 2009) that is used by several American Universities and school as an unofficial course guide.[3] In this system, the recommendation process is obtained by querying a relational database with course and student information. No specific recommendation approaches are used.

A hybrid approach is also adopted by RPL, the prototype system that has been implemented in the course repository of the Virtual University of Tunis.[4] This prototype includes a recommendation engine that combines a collaborative filtering algorithm with a content-based filtering algorithm, using data that has been logged and mined from user actions.

A more recent conceptual framework is ROLS (Recommender Online Learning System) (Peiris & Gallupe, 2012). It uses a collaborative approach combined with inferences made on a knowledge base composed of learning elements and procedure rules (specifying how each elements should be taught). The system is purposed at providing recommendations to learners based on content being delivered and assessed but also to provide data and feedback for all learning stakeholders (instructors, course developers and system administrators) to improve their experience with the system.

Table 3 compares the various available systems and prototypes together and with respect to our prototype tool. As it can be seen the greatest part of them uses a classical collaborative approach to recommendation and only few of them hybridize such approach with a more sophisticated one. Apart ROLS, our prototype is the only one that relies on knowledge structures (i.e. formally modeled domains of concepts) to provide better and more fine grained recommendations.

[3] http://www.courserank.com/.

[4] http://cours.uvt.rnu.tn/rpl/.

Table 3
Comparison of our tool with related work.

System	Status	Recommender approach	Sequencing capabilities	Recommendations based on
Altered vista	Full system	Collaborative	No	User ratings
QSIA	Full system	Collaborative	No	User ratings
CYCLADES	Full system	Collaborative	No	User ratings
ELS	Full system	Collaborative + clustering	No	User ratings
CourseRank	Full system	DB Filters	No	User ratings
RACOFI	Prototype	Collaborative + rules engine	No	User ratings
CoFind	Prototype	Collaborative	No	User ratings
Shen 04	Prototype	Content based	Yes	User knowledge
LSRS	Prototype	Markov chains	Yes	User knowledge
Re-mashed	Prototype	Collaborative	No	User ratings
RPL	Prototype	Collaborative + content based	No	User knowledge + user ratings
ROLS	Proof of concept prototype	Collaborative + knowledge structures	No	User knowledge + user ratings
OUR TOOL	Prototype (integrated in a commercial system)	Collaborative + knowledge structures	Yes	User knowledge

Our prototype is also one of the few prototypes (together with Shen, LSRS, RPL and ROLS) that bases recommendations not just on user ratings but also on user knowledge i.e. on past courses or on concepts that are considered as already known by the learner that is asking for recommendations. Moreover, like both Shen and LSRS, it is also able to provide sequencing capabilities i.e. the recommendation is not related to a course or to a learning resource but to a dynamically generated sequence of resources.

6. Experimentation and evaluation

6.1. Experimentation context and approach

To evaluate the prototype and analyze its effects in a learning process, we experimented it with real users within a University setting. In particular, 170 students enrolled in an online course on Software Engineering were involved in the experiment.

68 out of 170 students (40%) participated actively in the experience. We considered active participation the submission of an evaluation form at the end of the experience. Since the experiment was optional for all students, 60% of them chose not to send the evaluation form and thus they were excluded from the analysis.

From the 68 participants we formed 2 groups for the experiment. One experimental group with 41 students (60%) was enabled to the *formative needs* prototypal section 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 the control group composed of 27 students (40%). The formal assignment lasted three weeks during the second third of the Fall term. All students of both groups were supervised by a tutor during the experiments.

After the assignment, students of the experimental group were required to fill out a questionnaire that included the following 7 sections: (i) identification data; (ii) evaluation questions about the knowledge acquired within the course; (iii) open questions about the integrated system supporting the course; (iv) test-based evaluation of the recommendation features provided by the system; (v) test-based evaluation of the usability of f*ormative need* section (Ertl, Ebner, & Kikis-Papadakis, 2010). Students submitting this questionnaire had the chance to increase their final grade of the course up to 20%. If the questionnaire was not submitted or with wrong responses the final grade would not decrease whatsoever.

For those students of the control group, a different questionnaire was sent with only sections (i) and (ii) which had to be filled. Students submitting this questionnaire had the chance to increase their final grade of the course up to 10%. If the questionnaire was not submitted or with wrong responses the final grade will not decrease whatsoever.

For qualitative statistical analysis, we summarized the open answers in the surveys. For the quantitative statistical analysis we adopted basic statistics, such as Mean (M), Standard Deviation (SD) and median (Md). For the section (v) we used the System Usability Scale (SUS) Brooke, 1996.

From the editing point of view, the course was modeled within IWT by exploiting domain model discussed in Section 3.1. In order to provide recommendation facilities, 5 ULLG had been created to cover specific aspect of the *Software Requirements* subtopic (see Fig. 5).

The complete evaluation was presented in the context of the ALICE project[5] and also discussed in Capuano, Mangione, Pierri, and Salerno (2013). In the following we report the summary of the main findings concerned with the ULLG recommendation features integrated with the IWT system.

6.2. Evaluation results

The Fig. 6 shows the quantitative analysis in terms of Mean (*M*), Standard Deviation (SD) and Median (Md) related to the competence acquired with respect to specific concepts of the course covered by the defined ULLGs (*Software Requirements*). In general, we denote good levels of the acquired competences.

To evaluate differences between experimentation and control groups, the section (ii) of both questionnaires included an evaluative assignment with 2 questions about the *Software Requirements* topic as follows:

(1) From your experience as a user of social networks (e.g., Facebook, Twitter, etc.), indicate 5 functional requirements and 5 nonfunctional requirements implemented in these systems. Classify the nonfunctional requirements according to the Volere template.
(2) Indicate what the problems are to identify requirements during their elicitation.

While Question 1 is more general and practical Question 2 is more specific and theoretical. This aim was also to evaluate the impact of the prototype both on general and on specific acquisition of knowledge. This part of each questionnaire was assessed by a lecturer who used the standard 10-point scale to score the students' responses. Table 4 shows the results.

From the results of Table 4, students from the experimental group scored higher than the control group though the overall difference is not significant. However, observing closed the results, while Question 1 got similar marks, for Question 2 the marks were significantly different (1.22 out of 10). More interestingly, the SD in

[5] http://www.aliceproject.eu/.

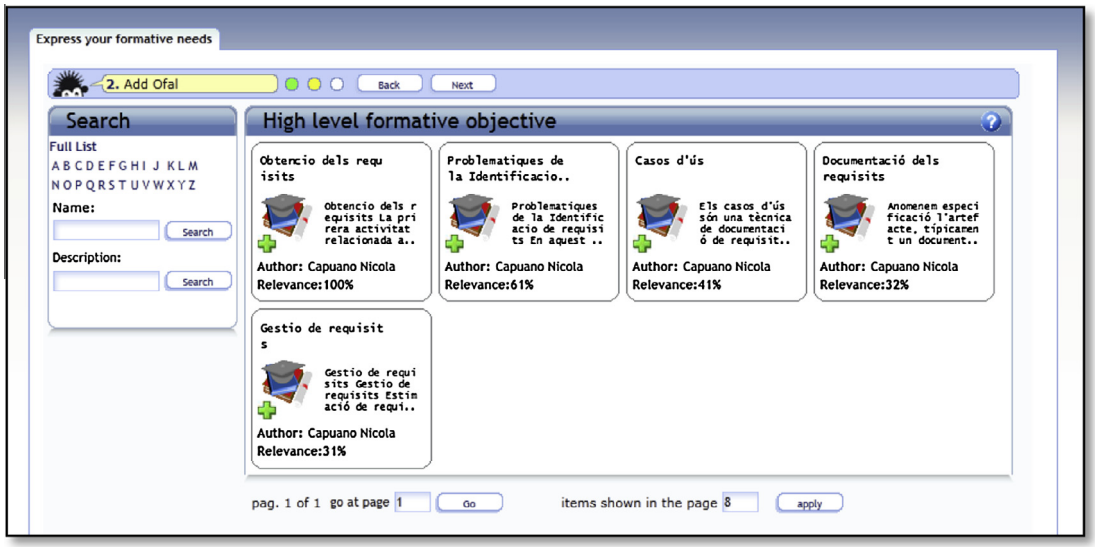

Fig. 5. The list of experimental ULLG.

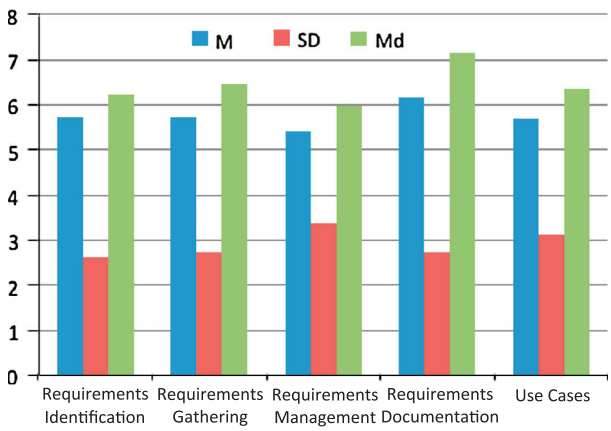

Fig. 6. Quantitative analysis of acquired competencies.

Question 2 of the experimental group is considerately lower than in the other group for the same question and also lower than the other questions and groups.

This result is in line with the fact that the students, by exploiting system recommendations, could find a specific resource devoted to answer this question while control group students had the information related to this question more dispersed in their material.

After having assessed the knowledge acquired by the students we have analyzed the value of integrated system as educational resource. To this end, quantitative and qualitative data were collected in sections (iii) and (iv) of the questionnaire by 3 open questions (qualitative) and 13 test-based questions (quantitative).

In the questionnaire, for the three quantitative questions we used a 0–10 rating scale. The scale went from the worst mark (0) to the best mark (10) considering a "good" assessment marks from 5 to 10 and a "bad" assessment marks from 0 to 4.9.

For the test-based questions the rating scale ranged from "Not at all" (1); "Somewhat" (2) and "Completely" (3). Despite sometimes these values changed to fit best the expected type of the responses, in all cases 3 options were provided (positive, medium and negative).

Three open questions were asked to students about the ULLG prototype:

(1) Evaluate in general the integrated system to support the study of the course on *Software Engineering* (and assess the system from this view in the scale 0–10).
(2) Indicate how in your opinion the integrated system has impacted in your individual learning process as for the *Software Engineering* topic (and assess the system from this view in the scale 0–10).
(3) In comparison to the standard system what advantages and disadvantages do you think the learning goals recommender prototype provides to study? Indicate in your view what are the main problems, issues and lacks of this tool (and assess the system from this view in the scale 0–10).

The average of 6.14 (SD = 2.27, Md = 7) has been achieved. This result is very good considering the prototype nature of the environment. In particular, students in general liked the platform and found it useful for their study (Question 1: $M = 6.13$, SD = 2.31, Md = 7). Question 2 was slightly better scored than Question 1 ($M = 6.39$, SD = 2.33, Md = 7). Finally, Question 3 was scored a bit lower that the other 2 questions, though not significantly ($M = 5.91$, SD = 2.19, Md = 6.5). Students commented that the integrated system provided a higher degree of flexibility and personalization with respect to the standard one.

Table 5 reports the answers provided to the 13 quantitative questions included in the section (iv) of the questionnaire for assessing the features of the integrated system. Also in this case results are encouraging with a predominance of answer of "completely" and "somewhat" types.

To investigate the overall usability of the IWT system, we used the SUS included in section (v) of the questionnaire. The answers were given on the 5-point Likert scale, so that students could state their level of agreement or disagreement. The rating scale ranged

Table 4
Results of the learning assignment evaluation.

Questions	Experimental group ($n = 41$)	Control group ($n = 27$)
Question 1	$M = 6.38$; SD = 1.64; Md = 6	$M = 6.11$; SD = 1.56; Md = 6
Question 2	$M = 7.83$; SD = 0.78; Md = 8	$M = 6.33$; SD = 1.28; Md = 6
Overall	$M = 7.11$; SD = 1.46; Md = 7	$M = 6.22$; SD = 1.41; Md = 6

Table 5
Evaluation of the platform added values.

	Questions	Completely	Somewhat	Not at all
1	The possibility to express your formative needs has allowed you to have more control over your learning?	8	21	12
2	Being able to express your needs in a simple language has contributed to motivate vour desire to learn?	11	16	14
3	Do you think that this solution of asking to take more responsibility over what you need helped you to capture a greater awareness on the right learning path?	16	13	12
4	Do you think that the answers obtained in terms of learning paths to follow by filling your needs are relevant and effective?	15	19	7
5	Do you think you can speed up learning time by eliminating states to which you are subject when your path is guided by the teacher?	14	14	13
6	The possibility to have specific learning path created *ad hoc* for filling your need has allowed you to obtain good results in terms of learning?	10	20	11
7	Did this learning modality have an impact on your participation in the learninq experience?	12	15	14
8	How you did you find the interaction with this new method of learning experience?	13	23	5
9	How quickly you adapted to this new method of expression through natural lanquaqev (veryfast, moderately, not very fast)?	9	23	9
10	Do you think that this new kind of interaction modality student-learning environment can be a step towards a self-regulated learning?	8	25	8
11	Do you think that the recommendations you received in terms of learning path to follow were tailored to your learning style and your profile?	10	20	10
12	How do you consider the learning path predisposed for you for filling your learning needs?	12	23	6
13	The ability to quickly obtain the recommendations brought you to express more than one need?	15	19	7

from "I strongly disagree" (1), "I disagree" (2), "neither/nor" (3) to "I agree" (4), "I strongly agree" (5).

SUS scores have a range of 0 to 100 with an average score of 68, obtained from 500 studies. A Score above a 68 would be considered above average and anything below 68 is below average. A score above an 80.3 is considered an A (the top 10% of scores). Scoring at the mean score of 68 gets you a C and anything below a 51 is an F (putting you in the bottom 15%).

After calculating the SUS score for each student, we got an average for 41 SUS scores of 60.78 thus below the SUS mean but nearby, which is a good score considering the first prototypical nature of the system.

7. Conclusions and future work

We defined in this paper a methodology to recommend learning goals and to generate learning experiences and a prototype component integrated in an commercial adaptive e-learning system named IWT. We compared the proposed approach with similar existing systems and prototypes facing the problem of learning resources and learning goals recommendation.

The first evaluation provided encouraging results considering the prototype nature of the environment. A more extensive experimentation is currently running in order to provide comments and suggestions to be used for models and methodologies improvement. In addition to comments coming from experimentation, some improvement can be already foreseen.

In particular the application of matrix factorisation techniques (Rendle & Schmidt-Thieme, 2008) able to transform the concept mapping matrix that is a huge sparse matrix in a product of smaller dense matrixes can be applied to optimize recommender performances. In addition, the possibility for learners to rate ULLGs created by other teachers or learners will be explored. This rating can be exploited by recommender algorithms as explicit feedback to improve recommendations.

Acknowledgements

The research reported in this paper is partially supported by the European Commission under the Collaborative Project ALICE "Adaptive Learning via an Intuitive, interactive, Collaborative, Emotional system", VII Framework Program, Theme ICT-2009.4.2, Grant Agreement n. 257639.

References

Acampora, G., Gaeta, M., Orciuoli, F., & Ritrovato, P. (2010). Exploiting semantic and social technologies for competency management. In *10th IEEE intl. conference on advanced learning technologies* (pp. 297–301).

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*(6), 734–749.

Albano, G., Gaeta, M., & Ritrovato, P. (2007). IWT: An innovative solution for AGS e-Learning model. *International Journal of Knowledge and Learning, Inderscience, 3*(2/3), 209–224.

Anderson, M., Ball, M., Boley, H., Greene, S., Howse, N., Lemire, D., & McGrath, S. (2003). RACOFI: A Rule-Applying Collaborative Filtering System. *IEEE/WIC COLA*.

Avancini, H., & Straccia, U. (2005). User recommendation for collaborative and personalised digital archives. *International Journal of Web Based Communities, 1*(2), 163–175.

Balabanovic, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommenddation. *Communications of the ACM, 40*(3), 66–72.

Bercovitz, B., Kaliszan, F., Koutrika, G., Liou, H., Zadeh, Z.M., Garcia-Molina, H. (2009). CourseRank: A social system for course planning. In *35th SIGMOD international conference on Management of data*.

Bodea, C. N., Dascalu, M. I., & Lytras, M. D. (2012). A recommender engine for advanced personalized feedback in e-Learning environments. *International Journal of Engineering Education, 28*(6), 1326–1333.

Brooke, J. (1996). *SUS: A "quick and dirty" usability scale*. Taylor and Francis: Usability Evaluation in Industry.

Burke, R. (2007). *Hybrid Web recommender systems lecture notes in computer science*. Berlin: Springer.

Capuano, N., Gaeta, M., Miranda, S., Orciuoli, F., & Ritrovato P. (2008). LIA: An intelligent advisor for e-learning. In *Lecture notes in computer science* (vol. 5288, pp. 187–196). Berlin: Springer-Verlag.

Capuano, N., Gaeta, M., Orciuoli, F., & Ritrovato, P. (2009). On-demand construction of personalized learning experiences using semantic web and Web 2.0 techniques. In *9th IEEE international conference on advanced learning technologies* (pp. 484–488).

Capuano, N., Gaeta, M., Orciuoli, F., & Ritrovato, P. (2010). Semantic Web Fostering Enterprise 2.0. In *4th International conference on complex, intelligent and software intensive systems* (pp. 1087–1092).

Capuano, N., Gaeta, M., Salerno, S., & Mangione, G.R. (2011). An ontology-based approach for context-aware e-Learning. In *3rd International conference on intelligent networking and collaborative systems* (pp. 789–794).

Capuano, N., Mangione, G. R., Pierri, A., & Salerno, S. in press. ALICE: Adaptive learning via interactive, collaborative and emotional approaches. In *Technological and Social Environments for Interactive Learning*. Informing Science Press.

Capuano, N., Gaeta, M., Ritrovato, P., & Salerno, S. (2008). How to integrate technology enhanced learning with business process management. *Journal of Knowledge Management, Emerald, 12*(6), 56–71.

Capuano, N., Mangione, G. R., Pierri, A., & Salerno, S. (2012). Learning Goals recommendation for self regulated learning. *International Journal of Engineering Education, 28*(6), 1373–1379.

Drachsler, H., Pecceu, D., Arts, T., Hutten, E., Rutledge, L., Van Rosmalen, P., Hummel, H.G.K., & Koper, R. (2009). ReMashed-recommendations for mash-up personal learning environments. In *Lecture Notes in Computer Science* (vol. 5794, pp. 788–793). Berlin: Springer-Verlag.

Dron, J., Mitchell, R., Boyne, C., & Siviter, P. (2000). CoFIND: Steps towards a self-organising learning environment. In *World conference on the WWW and internet* (pp. 146–151).

Ertl, B., Ebner, K., & Kikis-Papadakis, K. (2010). Evaluation of e-Learning. *International Journal of Knowledge Society Research, IGI-Global.* 1(3), 33–43.

Gaeta, M., Orciuoli, F., Paolozzi, S., & Ritrovato, P. (2009). Effective ontology management in virtual learning environments. *International Journal of Information and Enterprise Management, 6*(2), 96–123.

Gaeta, M., Orciuoli, F., & Ritrovato, P. (2009). Advanced ontology management system for personalised e-learning. *Journal of Knowledge-Based Systems, 22*(4), 292–301.

Huang, Y. M., Huang, T. C., Wang, K. T., & Hwang, W. Y. (2009). A markov-based recommendation model for exploring the transfer of learning on the web. *Educational Technology & Society, 12*(2), 144–162.

IMS Global Learning Consortium. (2006). *Best practice guide for IEEE 1484.12.1-2002 standard for learning object metadata.*

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). Grouplens: Applying collaborative filtering to usenet news. *Communications of ACM, 40*(3), 77–87.

Mangione, G. R., Gaeta, M., Orciuoli, F., & Salerno, S. A. (2010). Semantic metacognitive learning environment. cognitive and metacognitive educational systems. In *AAAI Fall Symposium*.

Peiris, K. D. A., & Gallupe, R. B. (2012). A conceptual framework for evolving recommender online learning systems. *Decision Sciences Journal of Innovative Education, 10*(3), 389–412.

Perugini, S., Goncalves, M. A., & Fox, E. A. (2004). Recommender systems research: A connection-centric survey. *Journal of Intelligent Information Systems, 23*(2), 107–143.

Rafaeli, S., Barak, M., Dan-Gur, Y., & Toch, E. (2004). QSIA-a Web-based environment for learning, assessing and knowledge sharing in communities. *Computers, Education, 43*(3), 273–289.

Recker, M. M., & Wiley, D. A. (2001). A non-authoritative educational metadata ontology for filtering and recommending learning objects. *Interactive Learning Environments, 9*(3), 255–271.

Rendle, S., & Schmidt-Thieme, L. (2008). Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In *The 2008 ACM conference on recommender systems.* ACM.

Shen, L., & Shen, R. (2004). Learning content recommendation service based-on simple sequencing specification. In *Lecture notes in computer science,* (vol. 3143, pp. 363–370). Berlin: Springer-Verlag.

Tang, T., & McCalla, G. (2003). Smart recommendation for an evolving e-Learning system. workshop on technologies for electronic documents for supporting learning. In *International conference on artificial intelligence in education*.