Project Number: **257639**

Project Title: ALICE: ADAPTIVE LEARNING VIA INTUITIVE/INTERACTIVE, COLLABORATIVE AND EMOTIONAL SYSTEMS

Instrument: Specific Targeted Research Projects

Thematic Priority: ICT-2009.4.2:Technology-Enhanced Learning

Project Start Date: June 1st, 2010

Duration of Project: 24 Months

Deliverable: **D7.2.2: Models and Methodologies for Upper Level Learning Goals Support v2**

Revision: 2.0

Workpackage: WP7: Adaptive Technologies for e-Learning Systems

Dissemination Level: Public

Due date: December 31st, 2011

Submission Date: December 31st, 2011

Responsible CRMPA

Contributors: N/A

PROJECT CO-FUNDED BY THE EUROPEAN COMMISSION WITHIN THE SEVENTH FRAMEWORK PROGRAMME (2007-2013)

Version History			
Version	**Date**	**Changes**	**Contributors**
0.1	30/11/2010	The deliverable structure has been completed, the introduction and the background chapters have been written. Sections 1 and 2 have been completed.	CRMPA
0.2	31/12/2010	A study about recommender systems and related approaches and algorithms has been performed. Section 3 has been completed. Section 4 is under development.	CRMPA
0.3	31/01/2011	Algorithms for conceptual mapping, concept utility estimation and ULLG Utility Estimation have been defined. The technological perspective has been outlined. Section 4 has been completed.	CRMPA
1.0	28/02/2011	Related algorithms and systems have been surveyed. Section 5 has been completed and Section 4 has been improved basing on the analysis of related work. Conclusions and future work have been written.	CRMPA
1.1	13/04/2011	Updated some formulas in section 4.3	CRMPA
1.2	30/09/2011	Related work improved with a comparison with similar systems. A subsection on latent factor models has been added to 3.2	CRMPA
1.3	30/11/2011	Extension of section 4.2 with the calculation of a cognitive component and the hybridization of such component with the collaborative one. An example of use was added as section 4.4.	CRMPA
2.0	20/12/2011	Technological perspective moved in section 5 and extended. Introductions and conclusions revised and extended. Some rewording was made following internal peer review.	CRMPA

Table of Contents

1 Introduction

Upper Level Learning Goals (ULLGs) are purposed to provide a high level access to the learning offer in order to simplify the learning courses building process. By exploiting ULLGs, the generation of a learning experience can start from the explicit or implicit request made by a learner in terms of needs to be satisfied (e.g. expressed in natural language) rather then from the selection of target concepts on an available domain model.

The purpose of this document is to provide the theoretical foundation for the management of ULLGs in the ALICE learning system with respect to requirements described in [21] (section 5.1). This will allow to improve and extend existing models, methodologies and components of ALICE reference platform IWT and to prepare it for a smooth integration of methodological and technological components coming from other ALICE research lines.

This document is structured in the following sections.

- **Section 2** provides an introduction about models and algorithms that are currently applied by IWT to manage ULLGs. In particular it currently supports two processes of course building starting from upper level learning goals, the first mapping an explicit request on pre-defined ULLGs, and the second mapping an explicit request directly on available domain models. This is a needed background to understand algorithms defined in section 4.
- **Section 3** provides an introduction on recommender systems i.e. systems able to provide personalized advice about the utility of items belonging to a given domain starting from the analysis of available information about users and items. Several kinds of approaches (cognitive, collaborative, hybrid) are presented and, for each of them, several techniques and algorithms are introduced. This section provides additional background to understand algorithms defined in section 4.
- **Section 4** defines improvements and extensions needed to IWT, from a theoretical perspective, to support new features basing on ULLGs. In particular a third process of course building starting from an implicit request rather than from an explicit one is introduced. Algorithms for concept mapping, concept utility estimation and ULLG utility estimation are provided basing on an hybrid recommending strategy combining a cognitive ontology-based approach with a collaborative approach that adapts and extends a user-to-user algorithm.
- **Section 5** describes the proposed improvements from a technological perspective by defining new software components to be developed and how they must be integrated in the existing IWT architecture. The section also describes the processes of learning goals selection and creation from the learner point of view.
- **Section 6** contextualizes performed research with respect to the relevant literature about recommender systems and their applications in technology enhanced learning

including existing systems and evaluation techniques. A comparison of our approach with similar systems is also provided as well as a set of techniques to evaluate the performances of recommender systems in general and in the technology enhanced learning domain.

- **Section 7** concludes the report and introduces next steps.

The document updates and extends [63]. In particular the concept utility estimation algorithm, part of the learning goals recommending process, has been strongly revised and updated by calculating a cognitive component based on the analysis of existing knowledge structures. This component is then hybridized with the already existing collaborative component in order to improve recommendations accuracy. An example of use of the defined algorithms has been introduced to demonstrate their effectiveness in a sample case.

The technological perspective has been improved and extended to include the new defined algorithms. New social based functions have been introduced with respect to ULLG sharing and rating. The ULLG selection and creation processes have been duly described from the learner point of view to guide components implementation. The document also integrates the analysis of latent factor models in the background section on recommender systems. It also extends the related work section with a comparison of our approach with similar systems.

2 Background

A significant educational action able to guide the learner along a comprehensive learning process is not only focused on learning (cognition level) but also on cultivating (in learners) a correct learning behaviour that empowers learners to achieve their learning goals in a controlled and directed way (metacognition level). To foster this aspect ALICE introduces the concept of ULLGs as a mean to simplify the access to the learning courses building process.

As reported in [1] and [2], the ALICE reference platform IWT and, specifically, its component LIA (Learning Intelligent Advisor) is able to build personalised *Units of Learning* (represented as sequences of *Learning Resources*) starting from a Learner Model and from a set of *Target Concepts* to be selected on a formally defined *Domain Model*. The figure 1 summarizes the standard process of Unit of Learning building as detailed in the section 2 of [1].

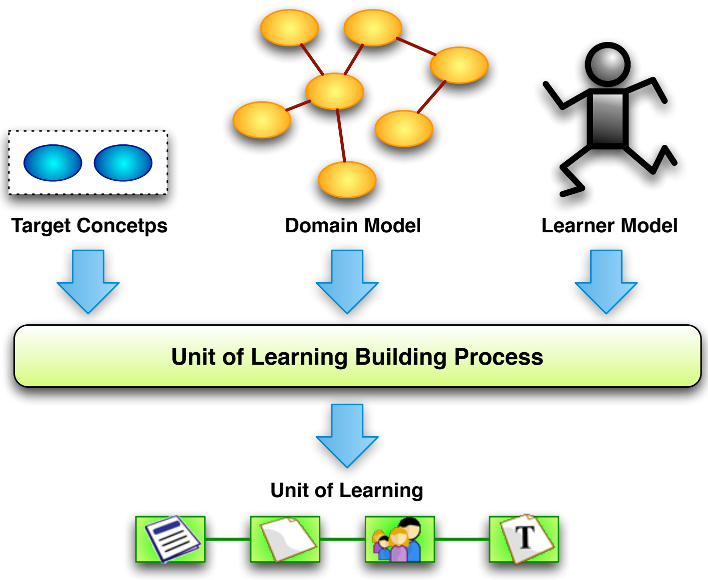

Figure 1. Input and output of the standard *Unit of Learning* building process.

In self-directed learning settings, this is translated into a need for the learner to deal with complex structures like the *Domain Model* (represented as a *Concept Graph* with additional didactical and contextual attributes for each node) in order to select feasible *Target Concepts* and let the system generate a personalized *Unit of Learning* for him.

To overcome this limitation and to simplify user interactions with the system, IWT already implements an alternative method for the expression of a learning need through *Upper Level Learning Goals (ULLG)*.

2.1 Upper Level Learning Goals

An *ULLG* is a meaningful set of *Target Concepts* on a given *Domain Model* with a connected textual description [22] [23]. *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 *ULLG*s 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 *Learner Model*. The figure 2 summarizes the process of *Unit of Learning* building exploiting *ULLGs*.

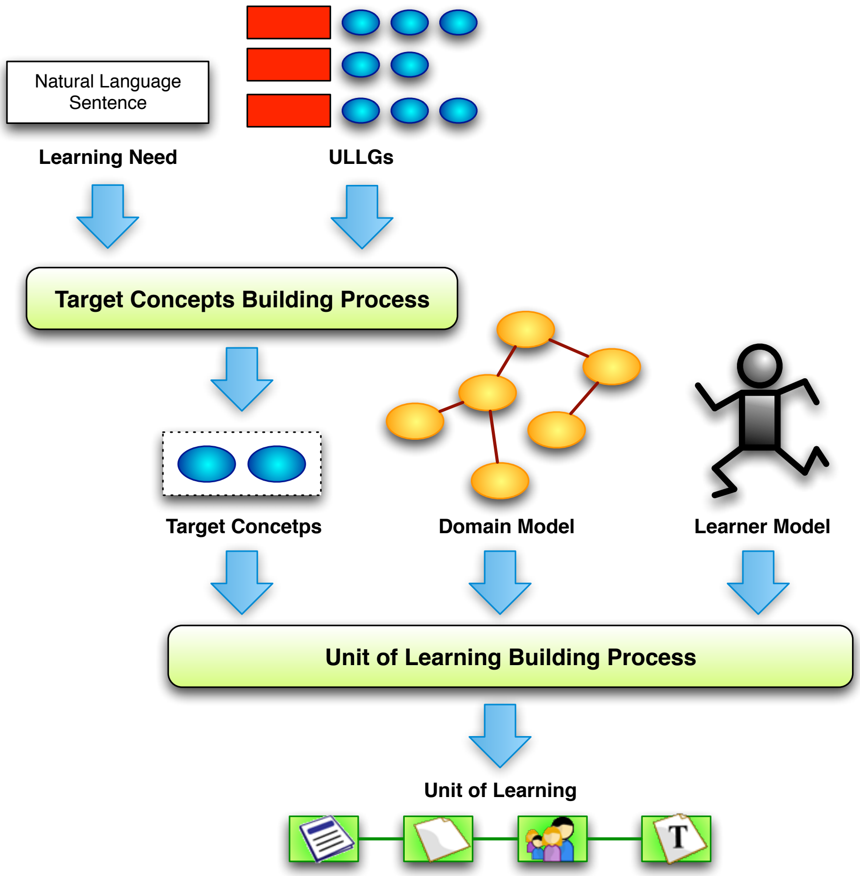

Figure 2. The revised *Unit of Learning* building process exploiting *ULLGs.*

The key addition with respect to the standard process (detailed in the section 2 of [1]) is the *Target Concept Building* Process that is described in the following paragraph.

2.2 Target Concepts Building Process

An *ULLG* can be defined as a tuple $ULLG_i = (D_i, TC_{i1}, \ldots, TC_{in})$ 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.

Once an *LN* is expressed by a learner, a sentence similarity algorithm is applied between *LN* and the D_i field of existing *ULLGs*. To do that *LN* and each D_i are transformed into vectors of terms pre-processed with *stemming* and *stop-word lists*. Terms coming from *LN* are enriched with synonyms coming from domain dictionaries automatically extracted from *Wikipedia*. For each $ULLG_i$, the similarity between D_i and *LN* is then calculated using the *Cosine Distance* and the *Levensthein Distance.*

The use of a lexical database enables to model human common sense knowledge and the incorporation of corpus statistics allows the method to be adaptable to different domains. An important aspect that this approach proposes (detailed in [3]) is that it takes care not only of the semantic similarities between the single words existing in the short sentences but also of the order of words within the phrase. This avoids that two sentences like “A quick brown dog jumps over the lazy fox” and “A quick brown fox jumps over the lazy dog” are considered exactly the same by the sentence similarity algorithm.

ULLGs presenting higher similarities with *LN* are provided to the learner as results. He can select one (or more) of them and request the building of the corresponding *Unit of Learning*. The figure 3 summarizes this process.

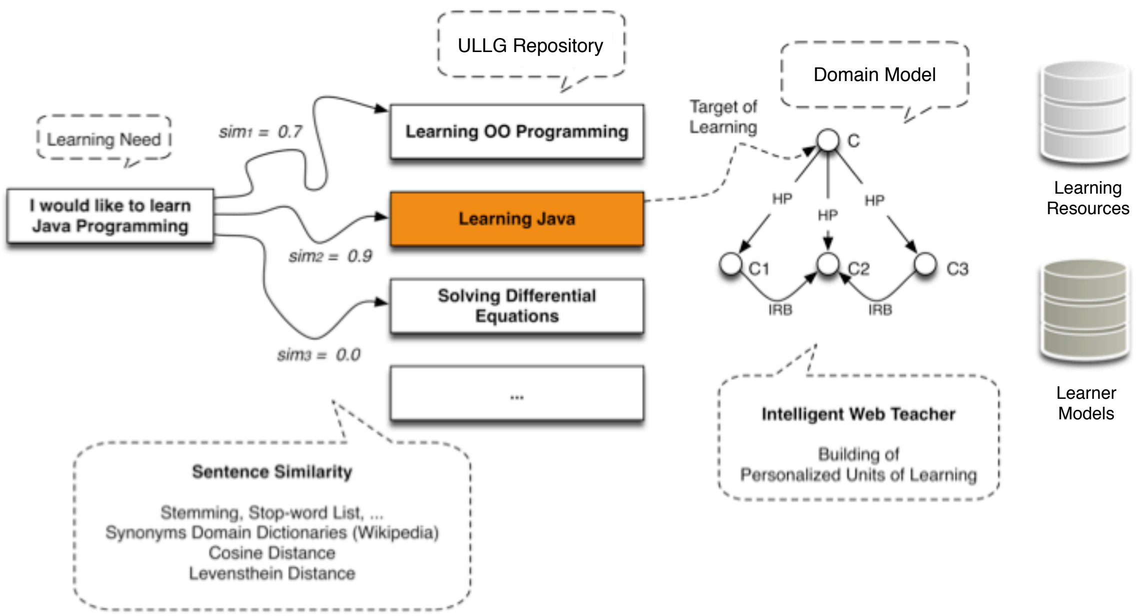

Figure 3. The processing of a Learning Need.

2.3 Alternative Process

In the case there is no *ULLG* in the repository satisfying the expressed *LN* then an alternative way to proceed is applied. This consists in matching the natural language sentence included in the *LN* directly with the concepts of the *Domain Model*.

First of all, the natural language sentence is analysed in order to extract pieces of knowledge and relevant concepts. The process consists in the application of a *stemming* algorithm to obtain the base form of words, an algorithm for *part of speech tagging* to obtain the syntactic category for each word and a *chunking and shallow parsing* algorithm to group words in noun phrases and simple verb phrases.

Then the *similarity relatedness* of extracted nouns and verbs with the concepts of available *Domain Models* is calculated using Wu & Palmer similarity measure and *Synonyms Domain Dictionaries* [4]. Concepts with high similarity to the natural language sentence are presented to the learner and he can select one or more of them as *Target Concepts* to start the definition and the execution of a new personalized e-learning experience. The process is summarized in figure 4.

This process allows the learner to define his/her learning needs and direct his/her learning experience (goal setting), to explore the conceptual space developing a larger locus of control and to determine when individual goals have been adequately addressed (self control).

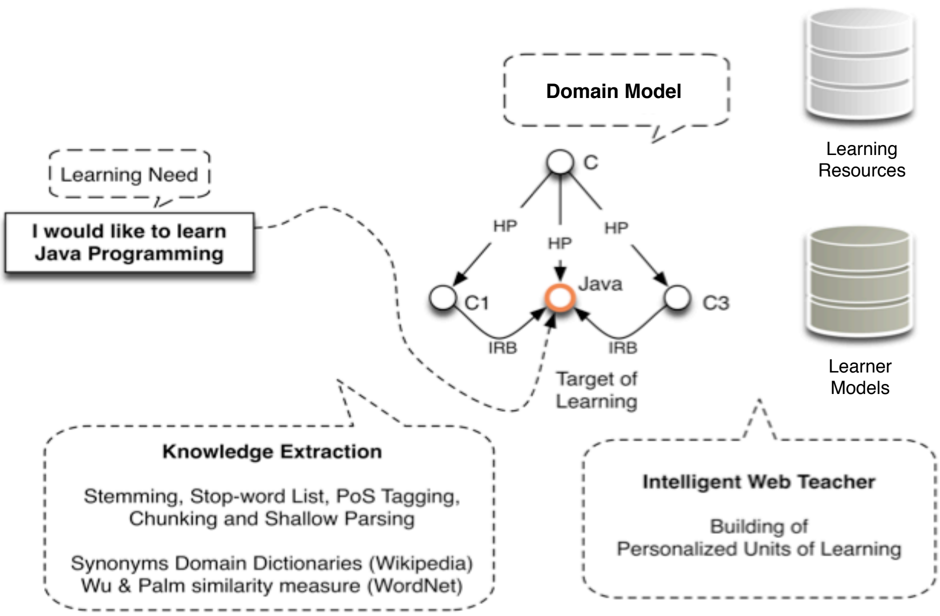

Figure 4. The alternative Target Concept Building Process.

3 Recommender Systems

Recommender Systems (RS) are purposed to give users 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 [5]: 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 that maximizes the utility function so that:

$$i'_c = \underset{i \in I}{\arg\max}\, u(c,i). \tag{1}$$

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 preferences on each object of the available catalog. 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.

The techniques, by which it is possible to predict unknown ratings from those notes, are a fundamental aspect that allows for characterizing such systems. In particular there are three broad categories of approaches to recommendations in the literature:

- **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 the following, we will investigate the three approaches by considering the advantages and disadvantages of each one of them.

3.1 Cognitive Approaches

In cognitive approaches [6], 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. For example, in an application for movies recommendation, the system would try to understand the similarities between the movies that the user has positively rated in the past and those currently available (e.g. same genre, same director, common actors, etc.). After that, only objects with high similarity would be selected and proposed to the user.

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) = ($w_{i,1}$, … $w_{i,k}$) 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 weight of the considered attributes can be created automatically by the system (e.g. the frequency of keywords in text-based objects) or manually by a user (e.g. the presence or absence of a specific tag associated with the object).

As for the objects, users are also associated with a profile based on the attributes of the objects preferred in the past. This profile is defined as *profile*(c) = ($w_{c,1}$, … $w_{c,k}$), 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 the 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:

$$sim(profile(c), content(i)) = \frac{\sum_{j=1}^{k} w_{c,j} w_{i,j}}{\sqrt{\sum_{j=1}^{k} w_{c,j}^2}\sqrt{\sum_{j=1}^{k} w_{i,j}^2}}. \tag{2}$$

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. Basing only on the user's past history, in fact, the recommendations tend to follow his preferences too closely and do not allow serendipity (the chance to discover useful things even if they differ from one’s preferences). This can lead sometimes to consider the system useless, given the obviousness of its suggestions.

3.2 Collaborative Approaches

In collaborative approaches [7], unknown values of the utility function $u(c, i)$ are estimated from those made available by people considered similar to c. 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.

In particular, there are *memory-based* algorithms (based on the history of the evaluations of system users to predict future evaluations) and *model-based* algorithms (which do not use the history of the system to make predictions but use it to learn a model that is then used to generate recommendations). Among the former class, the most popular are *user-to-user* and *item-to-item* algorithms discussed below. In the second class, instead, one of the emerging approaches is the one based on *latent factor models* discussed in 3.2.3.

3.2.1 User-to-User Algorithms

U*ser-to-user* algorithms [8] calculate the utility $u(c, i)$ as aggregation of the utility expressed for i by users similar to c; in other words:

$$u(c,i) = \underset{c' \in C'}{aggr}\, u(c',i) \tag{3}$$

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:

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

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*. These measures are applied to the vectors ($w_{c,1}$, … $w_{c,m}$) 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. Although these algorithms are widely used, they also have some limitations. The main problem occurs in domains characterized by 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. A less central aspect is the problem that afflicts the very common objects in the catalog or that are commonly preferred by a wide range of users. This leads to ever recommend those objects for all users.

3.2.2 Item-to-Item Algorithms

A variant of the *user-to-user* algorithm is the *item-to-item* recommendation algorithm [9] that was created to address the *new user* problem in environments where it is necessary to provide fast and accurate recommendations to those who have just joined the system. The algorithms in this category compute the utility $u(c, i)$ as aggregation of the utility expressed by c for objects similar to I, or:

$$u(c,i) = \underset{i' \in I'}{aggr}\, u(c,i') \tag{5}$$

where I' is the set of the m objects considered most similar to i (with m chosen between 1 and the number of available objects). The simplest aggregation function is the average of the ratings given by c to the objects of I', possibly weighted on the degree of similarity with a formula similar to (4).

The similarity between two objects is computed using the aforementioned measures like the *cosine similarity* (2) or the *Pearson's correlation coefficient*. These measures are applied to the vectors ($w_{i,1}$, … $w_{i,n}$) which characterize the objects where $w_{i,c} = u(c, i)$, if defined. Once the correlations between all pairs of objects have been calculated, the value of the utility can be calculated with a formula similar to (4):

$$u(c,i) = \frac{\sum_{i' \in I'} u(c,i') \cdot sim(i,i')}{\sum_{i' \in I'} \left| sim(i,i') \right|} \tag{6}$$

This approach can provide fairly accurate recommendations also to users who have rated only one object in the catalog. It is therefore useful in systems with many users and/or objects and when the number of available ratings is low. Unfortunately the approach suffers from the same limitations of user-to-user algorithms with the difference that it partially solves the *new user* problem.

3.2.3 Latent Factor Models

In contrast to *memory-based* approaches, techniques for *model-based* recommendation [10] do not directly use the history of the system to make predictions but use it to learn a model that is then used to generate recommendations. This category includes systems that use Bayesian networks, neural networks and clustering techniques to represent the problem.

These approaches, by creating a data model from which to infer domain properties useful to recommendations allow, in general, to achieve more accurate results than *memory-based* methods reported in 3.2.1 and 3.2.2. For this reason, in areas where the precision is critical, *model-based* systems may be the best solution, although they should renounce to the simplicity of the competitor algorithms.

Among these approaches, *latent factor models* try to explain the ratings by characterizing items and users on a small set of factors inferred from the rating patterns. Such factors comprise a computerized alternative to human-created attributes discussed in 3.1. Some of the most successful implementations of this approach are based on *matrix factorization*.

In its basic form, *matrix factorization* characterizes both items and users by vectors of factors inferred from item rating patterns. High correspondence between item and user factors leads to a recommendation. These methods have become popular in recent years by combining good scalability with predictive accuracy. In addition, they offer much flexibility for modeling various real-life situations [64].

Matrix factorization models map users and items to a latent factor space of dimensionality *f*. In other words, each item *i* is associated with a vector $q_i \in R^f$, while each user *c* is associated with a vector $p_c \in R^f$. For a given item *i*, the elements of q_i measure the extent to which the item possesses those factors, positive or negative. For a given user *c*, the elements of p_c measure the extent of interest the user has in items that are high on the corresponding factors, again, positive or negative.

In this space, the dot product between q_i and p_c captures the interaction between the user *c* and the item *i* representing the user's overall interest in the item's characteristics. The utility function $u(c, i)$ can be so approximated in this way:

$$u(c, i) = q_i^T p_c \tag{7}$$

The major challenge of this approach is computing the mapping of each item and user to the factor vectors q_i and p_c. A common approach here is to minimize the regularized squared error on the set of known ratings [65][66] i.e. to calculate:

$$\min_{q,p} \sum_{(c,i)\in\kappa} \left(u(c,i) - q_i^T p_c\right)^2 + \lambda\left(\|q_1\|^2 + \|p_c\|^2\right) \tag{8}$$

where κ is the set of the (c, i) pairs for which $u(c, i)$ is known (i.e. items explicitly or implicitly rated by the users) and the constant λ controls the extent of regularization and is usually determined by cross-validation. Several algorithms have been proposed to solve this optimization problem. The most popular are the stochastic gradient descent [67] and ALS (Alternating Least Square) techniques [68].

Moreover, the matrix factorization approach lends itself well to modeling temporal effects, which can significantly improve accuracy. In real applications, in fact, items perception and popularity constantly change as new selections emerge. Similarly, users' inclinations evolve, leading them to redefine their taste. Several approaches [69][70] have been proposed to take into account the temporal effects taking the user factors as a function of time.

3.3 Hybrid Approaches

Hybrid approaches [11] try to overcome problems of cognitive and collaborative approaches by using the two techniques simultaneously. The cognitive algorithms provide acceptable recommendations, also in cases where the data is minimal, while the collaborative algorithms can address the need to generate not only obvious but interesting recommendations.

There are several methods by which collaborative and cognitive approaches may be combined into a single system. Among them we quote the following [12]:

- *weighted hybridization* (a cognitive and a collaborative algorithms are developed and, as final result, a combination of predictions from the two approaches 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);
- *joint hybridization* (recommendations from all the available algorithms are presented to the user);
- *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* (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.

4 A Recommender System for ULLG

As we have seen in section 2, IWT supports two processes of course building starting from upper level learning goals, the first mapping an explicit request on pre-defined ULLGs, the second mapping an explicit request directly on available domain models. This document deals with the integration in IWT of a third process of course building 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 this cognitive state with cognitive states of similar learners and, from the other side, on the analysis of domain models organizing concepts belonging to ULLGs.

To do that we adopt a hybrid recommending strategy combining a cognitive ontology-based approach with a collaborative approach that adapts and extends a user-to-user algorithm (see 3.1.2). The proposed recommender algorithm consists of the following steps.

- Concept mapping: for each learner, known concepts plus concepts currently under learning (i.e. part of units of learning the learner is enrolled in) are identified.
- Concept utility estimation: for each learner, the utility of each unknown concept is estimated by looking at domain models and at concepts known and under learning by similar users (i.e. by users with similar concept mappings).
- ULLG utility estimation: the utility of each available ULLG is calculated for each learner by aggregating utilities of composing concepts.

Once the utility of each ULLG is estimated for a learner, the ULLGs with the greatest utility can be suggested to him. The proposed methodology upholds the social presence while supporting the development of self-regulated learning. Educational recommendations serves as a pedagogical advance organizer for the learners' community, as it anticipates and spreads needs, knowledge and learning paths. Furthermore it also supports help seeking processes improving the students' control over learning.

The following paragraphs deal with the description of each of these steps also providing a usage sample in a simulated learning environment to better understand explained concepts.

4.1 Concept Mapping

Given a set of concepts C and a set of learners L the *cognitive state* of a learner $l \in L$, as specified in [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 then 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 specified in [1], 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, …, 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 he is enrolled in. Then we can define the **concept mapping function** *CMF*: $L \times C \rightarrow [0, 1]$ as follows:

$$CMF(l,c) = \begin{cases} 1 & \text{if } CS_l(c) \geq \theta \\ 1/2 & \text{if } CS_l(c) < \theta \wedge c \in COT_l \\ 0 & otherwise \end{cases} \quad (9)$$

So, given a leaner, the CMF is 1 for all concepts that are already known by him and 0,5 for all concepts that are currently under learning. It is 0 for other concepts. The concept mapping function so represents an implicit rating given by a learner to available concepts: concepts that are relevant for him because learnt or under leaning are positively evaluated while for other concepts the evaluation is 0.

4.2 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 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. In particular:

- the *cognitive component* suggests concepts that are ontologically complementary to those already known or under learning;
- the *collaborative component* suggests concepts under learning or already known by similar users.

The following subsections describe how these components are calculated and hybridized to build a unique estimation of the concept utility.

4.2.1 Cognitive Component Estimation

As specified in [1], a *domain model*, describes the knowledge object of teaching through a set of concepts and a set of relations between concepts. Given a domain d, we can then represent the set of concepts belonging to d as C_d. We can then define the cognitive utility of a generic concept belonging to C_d for the learner l in this way:

$$u_{cog}(l,C_d) = \frac{\sum_{c \in C_d} CMF(l,c)}{|C_d|}. \tag{10}$$

The value of $u_{cog}(l, C_d)$ is so proportional to the number of concepts *l* has acquired in the domain *d* normalized on the total number of concepts belonging to the same domain. This value can be then directly used to represent the cognitive component of the concept utility of a single concept *c* for the learner *l* by settling:

$$u_{cog}(l,c) = u_{cog}(l,C_d) \mid c \in C_d. \tag{11}$$

In the previous equation it is important to specify that the set C_d can be uniquely determined given that any concept must belong to exactly one domain.

4.2.2 Collaborative Component Estimation

By adapting what explained in 3.2.1, 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*. In our case this will constitute the collaborative component of the concept utility $u_{coll}(l,c)$. The estimation can be done through the following formula obtained by adapting (4):

$$u_{coll}(l,c) = \frac{\sum_{l' \in L'} CMF(l',c) \cdot sim(l,l')}{\sum_{l' \in L'} |sim(l,l')|} \tag{12}$$

where *L'* is the set of the *n* most similar learners to *l* while *sim*(*l*,*l'*) is the similarity degree between the learner *l* and the learner *l'* obtained though similarity measures like the *cosine similarity* or the *Pearson correlation coefficient* (defined in section 3) calculated on *CMF*.

From the algorithmic point of view, in order to estimate the collaborative component of the concept utility function, we start from the concept mapping matrix where each element *CMF*(*l*,*c*) is defined with (9). This matrix is built the first time by considering every cognitive state and every course available on the system. Each time a learner enrols or abandons a course and after each testing activity, the row corresponding to this learner is updated, again, through equation (9).

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 $u_{coll}(l,c)$ for a given learner *l*, it is necessary to isolate and combine by applying equation (12) the utility expressed for *c* by the *n* learners more similar to *l* with *n* empirically defined.

Latent factor models presented in section 3.2.3 can be used to improve performances in the estimation of the collaborative component of the concept utility especially when the number of learners and of available concepts increase. In this case, instead of maintaining the huge utility matrix it is more performing to factorize such matrix and obtain it as a product of two smaller matrices. The equation (12) can be substituted by the equation (7) and, to determine q_i and p_c components, gradient descent or ALS techniques can be used.

4.2.3 Hybridization

After having calculated its cognitive and collaborative components, the utility of a concept *c* for a learner *l* can be estimated by composing these two values by also remembering to put it equal to 0 for concepts already known or that will be known soon according to the concept mapping function:

$$u(l,c) = \begin{cases} \alpha\ u_{cog}(l,c) + (1-\alpha)\ u_{coll}(l,c) & \text{if } CMF(l,c)=0 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

where α 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). The choice for α will be done empirically basing on experimentation results. Low values for α privileges serendipity while high values privilege accuracy of suggestions given by the recommender system.

4.3 ULLG Utility Estimation

The section 2.2.1 of [1] explains 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 existing Upper Level Learning Goal $ULLG_i$, the corresponding learning path $LPath_i$. Once determined learning paths associated to ULLGs, it is possible to estimate the **conceptual utility** $u_{con}(l,ULLG_i)$ of each of them for a learner *l* with the following equation:

$$u_{con}(l,ULLG_i) = \sum_{c \in LPath_i} \frac{u(l,c)}{|LPath_i|}. \tag{14}$$

The calculus of the conceptual 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 conceptual 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** $u_{mar}(l,ULLG_i)$ of $ULLG_i$ for a learner *l* that can be obtained with the following equation:

$$u_{mar}(l,ULLG_i) = \sum_{c \in LPath_i} \frac{u(l,c)}{\left|\{c' \in LPath_i \mid CMF(l,c')=0\}\right|}. \tag{15}$$

The marginal utility only takes into account the utility of concepts that the learner doesn't know yet and that are not included any of the courses he is enrolled in. 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 equation:

$$u(l,ULLG_i) = \beta\ u_{con}(l,ULLG_i) + (1-\beta)\ u_{mar}(l,ULLG_i). \tag{16}$$

where β is the hybridisation 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 β will be done empirically basing on experimentation results. Low values for β privileges accuracy while high values privilege novelty of suggestions given by the recommender system.

The integration of cognitive and collaborative components allows to mitigate the **cold start** problem discussed in 3.2.1 in fact the cognitive component can give useful recommendations also when the available quantity of data is very poor. In order to obtain some results also when no data at all is available about the current user, a *baseline recommender* can be added to the process.

The **baseline recommender** is activated only when the system has no suggestions for a given user i.e. when the utility of every ULLG is equal to zero. This happen for new learners i.e. learners that have an empty cognitive state and are not enrolled in any course. In such case the average social rating associated to each ULLG (see section 5.1) is used as an estimation of the popularity i.e. the average perceived utility of this ULLG. In other words, if $u(l, ULLG_i) = 0$ for any available $ULLG_i$ then:

$$u(l,ULLG_i) = \frac{r(ULLG_i)}{10}. \tag{17}$$

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

4.4 Example of Use

In order to show how the algorithms described in the previous subsections work in practice we present here an example of use of the system in a simplified context. Let's suppose to work in a learning environment with three domains *A*, *B* and *C* organizing a total of 10 concepts c_0, …, c_9 as shown in table 1 (where c_0, c_1, c_2 and c_3 belongs to *A*, c_4, c_5, c_6 and c_7 belongs to *B* while c_8 and c_9 belongs to *C*).

Domain Models										
Concept	0	1	2	3	4	5	6	7	8	9
Domain	A	A	A	A	B	B	B	B	C	C

Table 1. The sample domain models.

Let's suppose that 7 learning goals $ULLG_0$, …, $ULLG_6$ have been defined on these concepts as shown in table 2 (where c_0, c_1, c_2 and c_3 belongs to $ULLG_0$; c_0 and c_1 belongs to $ULLG_1$; c_2 and c_3 belongs to $ULLG_2$; c_4, c_5, c_6 and c_7 belongs to $ULLG_3$; c_4 and c_5 belongs to $ULLG_4$; c_6 and c_7, belongs to $ULLG_5$; c_8 and c_9 belongs to $ULLG_6$).

In this learning environment we have 5 users u_0, …, u_4. Each of them already knows some concept and is currently studying other concepts as reported in the concept mapping table represented in table 3 and obtained applying the equation (9).

		Concepts									
		0	1	2	3	4	5	6	7	8	9
ULLGs	0	1	1	1	1						
	1	1	1								
	2			1	1						
	3					1	1	1	1		
	4					1	1				
	5							1	1		
	6									1	1

Table 2. The sample Upper Level Learning Goals.

		Concepts									
		0	1	2	3	4	5	6	7	8	9
Users	0	1,00	1,00			0,50	0,50	0,50			
	1	1,00	1,00	1,00	1,00					1,00	1,00
	2					1,00	1,00	1,00		0,50	0,50
	3								1,00	1,00	1,00
	4	1,00	1,00	1,00	1,00						

Table 3. The Concept Mapping Table.

As it can be seen u_0 knows c_0 and c_1 while he is enrolled in a course on c_4, c_5 and c_6; u_1 knows c_0, c_1, c_2, c_3, c_8 and c_9; u_2 knows c_4, c_5 and c_6 while he is enrolled in a course on c_8 and c_9; u_3 knows c_7, c_8 and c_9; u_4 knows c_0, c_1, c_2 and c_3. Let's start estimating the utility of each available concept starting from the **cognitive component**. The table 4 shows the $u_{cog}(l, c)$ for any learner l and concept c obtained applying equations (10) and (11).

		Concepts									
		0	1	2	3	4	5	6	7	8	9
Users	0	0,50	0,50	0,50	0,50	0,38	0,38	0,38	0,38	0,00	0,00
	1	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	1,00	1,00
	2	0,00	0,00	0,00	0,00	0,75	0,75	0,75	0,75	0,50	0,50
	3	0,00	0,00	0,00	0,00	0,25	0,25	0,25	0,25	1,00	1,00
	4	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00

Table 4. Cognitive Component of the Concept Utility.

The table 4 shows that the utility of a concept increases when the number of known concepts of the same domain increases. As an example the utility of concepts c_2 and c_3 for u_0 is 0,50 because he already knows ½ of the concepts of the same domain while the utility of c_8 and c_9 for the same user is 0 because no concepts are currently known by him in this domain.

In the same way, the utility of concept c_7 for u_2 is 0,75 because ¾ of the concepts of the same domain are known by him. Concepts under study are weighted ½ in this computation so the utility of c_7 is 0,38 for u_0 because he is already studying ¾ of the concepts of the same domain (¾·½ ≅ 0,38).

To calculate the **collaborative component** of the concept utility, first of all, the user-to-user similarity matrix is calculated on the concept mapping table (table 3) by applying the cosine similarity function defined by equation (2). The resulting matrix is shown in table 5.

		Users				
		0	1	2	3	4
Users	0	1,00	0,49	0,48	0,00	0,60
	1	0,49	1,00	0,22	0,47	0,82
	2	0,48	0,22	1,00	0,31	0,00
	3	0,00	0,47	0,31	1,00	0,00
	4	0,60	0,82	0,00	0,00	1,00

Table 5. User-to-user similarity matrix.

As it can be seen, most similar users are u_1 and u_4 followed by u_0 and u_4 while users u_0 and u_3; u_2 and u_4; u_3 and u_4 have no similarities at all. Basing on the defined similarity matrix, the collaborative component can be calculated by applying the equation (12). The table 6 shows the obtained $u_{coll}(l, c)$ for any learner l and concept c.

		Concepts									
		0	1	2	3	4	5	6	7	8	9
Users	0	0,69	0,69	0,69	0,69	0,31	0,31	0,31	0,00	0,46	0,46
	1	0,65	0,65	0,41	0,41	0,23	0,23	0,23	0,24	0,29	0,29
	2	0,69	0,69	0,22	0,22	0,24	0,24	0,24	0,31	0,52	0,52
	3	0,60	0,60	0,60	0,60	0,40	0,40	0,40	0,00	0,80	0,80
	4	1,00	1,00	0,58	0,58	0,21	0,21	0,21	0,00	0,58	0,58

Table 6. Collaborative Component of the Concept Utility.

The table 6 shows that the utility of concepts c_2 and c_3 is very high (0,69) for user u_0 because such concepts are already known by his most similar user (u_1). For the same reason, concepts u_8 and u_9 also have an high utility (0,46). Instead, the utility of concept c_7 for u_0 is 0 because no similar users know (or are currently studying) it.

By hybridizing cognitive and collaborative components through equation (13) it is possible to calculate the overall concept utility $u(l, c)$ for any learner l and concept c. This is reported in table 7 where the hybridisation coefficient α of (13) is settled to ½. As foreseen by the second condition of (13), $u(l,c)$ is settled to 0 for any couple (l,c) so that $CMF(l,c) > 0$ i.e. the utility of a known concept or of a concept under study is zero.

		Concepts									
		0	1	2	3	4	5	6	7	8	9
Users	0			0,60	0,60				0,19	0,23	0,23
	1					0,12	0,12	0,12	0,12		
	2	0,35	0,35	0,11	0,11				0,53		
	3	0,30	0,30	0,30	0,30	0,32	0,32	0,32			
	4					0,11	0,11	0,11		0,29	0,29

Table 7. The Concept Utility.

As it can be seen from table 7, the greatest utility for the user u_0 is represented by concepts c_2 and c_3. This is because these concepts not only complete his knowledge about the domain *A* but also correspond to concepts known by users similar to him (u_1 and u_4). Concepts c_8 and c_9 have a lower utility for u_0 because, even if they correspond to concepts known by the similar user u_1, they belong to a new domain.

The second greater figure in table 7 is represented by the utility of the concept c_7 for the user u_2. Also in this case the high value is due to both cognitive and collaborative components. c_7 in fact, from one side complete the user knowledge of domain *B* and, from the other side, is known by the similar user u_3. Also concepts c_0 and c_1 have an high utility for u_2 because they are known by two similar users (i.e. u_0 and u_1) even if they belong to a new domain.

Once the utility is calculated for each user and each concept, such utility must be aggregated to determine the utility of each available ULLG (as reported in table 2). To do that the system firstly calculate the marginal utility u_{con} by applying the equation (14). The results of this calculation are reported in table 8.

		ULLGs						
		0	1	2	3	4	5	6
Users	0	0,30	0,00	0,60	0,05	0,00	0,09	0,23
	1	0,00	0,00	0,00	0,12	0,12	0,12	0,00
	2	0,23	0,35	0,11	0,13	0,00	0,26	0,00
	3	0,30	0,30	0,30	0,24	0,32	0,16	0,00
	4	0,00	0,00	0,00	0,08	0,11	0,05	0,29

Table 8. The Conceptual Component of the ULLG Utility.

Such component is obtained by summing, for each ULLG, the utility of member concepts for a given learner and by dividing this number by the total number of concepts composing the ULLG. As an example the conceptual utility of $ULLG_0$ for u_0 is 0,30 while the utility of $ULLG_2$ for the same user is 0,60. Both ULLGs include concepts c_2 and c_3 that have a concept utility of 0,60 for u_0 but $ULLG_0$ includes a total of 4 concepts so its utility is ½·0,60 while $ULLG_2$ only includes c_2 and c_3 so its utility is exactly 0,60.

The calculus of the conceptual utility takes into account the utility of all concepts explained by the ULLG. This means that, if the ULLG includes many concepts already known by the learner, its conceptual utility can be low even if the utility of remaining concepts is high. The marginal component of the ULLG utility u_{mar} is purposed to rebalance this value through equation (15). The result of this calculation on our sample is reported in table 9.

		ULLGs						
		0	1	2	3	4	5	6
Users	0	0,60	0,00	0,60	0,19	0,00	0,19	0,23
	1	0,00	0,00	0,00	0,12	0,12	0,12	0,00
	2	0,23	0,35	0,11	0,53	0,00	0,53	0,00
	3	0,30	0,30	0,30	0,32	0,32	0,32	0,00
	4	0,00	0,00	0,00	0,08	0,11	0,05	0,29

Table 8. The Marginal Component of the ULLG Utility.

The marginal component is obtained by summing, for each ULLG, the utility of member concepts for a given learner and by dividing this number only by the total number of concepts composing the ULLG that are not known or under learning by the user.

As an example, the conceptual utility of both $ULLG_0$ and $ULLG_2$ for u_0 is the same because both include the same unknown concepts despite that the first includes a greater number of concepts at all. This is the same regarding $ULLG_3$ and $ULLG_5$ for u_0 and, in general for ULLGs that include the same subset of unknown concepts. By combining both components through equation (16) it is possible to estimate the whole utility of available ULLGs. This calculation is reported in table 9 where the hybridisation coefficient β of (16) is settled to ½.

		ULLGs						
		0	1	2	3	4	5	6
Users	0	0,45		0,60	0,12		0,14	0,23
	1				0,12	0,12	0,12	
	2	0,23	0,35	0,11	0,33		0,40	
	3	0,30	0,30	0,30	0,28	0,32	0,24	
	4				0,08	0,11	0,05	0,29

Table 9. The ULLG Utility.

Basing on these utility values, suggestions can be made to system learners. In particular, the ULLGs corresponding with the n greater utilities are suggested to each user. In our case, by settling n=3 $ULLG_2$, $ULLG_0$ and $ULLG_6$ are suggested to u_0; $ULLG_3$, $ULLG_4$ and $ULLG_5$ are suggested to u_1; $ULLG_5$, $ULLG_1$ and $ULLG_3$ are suggested to u_2; $ULLG_4$, $ULLG_0$ and $ULLG_1$ are suggested to u_3; $ULLG_6$, $ULLG_4$ and $ULLG_3$ are suggested to u_4.

5 Technological Perspective

Chapter 4 of [1] already presents the IWT logical architecture divided in the following layers:

- *Framework* used by developers to design and implement core services, application services and learning applications;
- *Core Services* providing basic features like resources management, ontology storing, user authentication, content storing, metadata, role and membership management, learning customisation, logging, profiling etc.
- *Application Services* used as building blocks to compose e-learning applications for specific domains including document management, conferencing, authoring, learning management, learning content management, ontology management, communication and collaboration, ULLG management, process management and information search services.
- *Learning Applications* covering specific learning scenarios obtained as integration of application services.

In the following subparagraphs we will present the extensions needed to IWT to implement and integrate defined algorithms from two complementary points of view: the architecture and the user experience.

5.1 Extensions Needed to IWT

From the technological point of view the recommender system for ULLG will be implemented as an extension of the already existing ULLG manager. Currently this component is made of the following modules:

- the *ULLG Designer* purposed to define an ULLG as aggregation of target concepts, a metadata and a text describing the ULLG in a natural language;
- the *ULLG Selector* that implements the target concept building process described in 2.2 to find the best ULLG starting from a learner query in a natural language;
- the *Concepts Selector* that implements the alternative process described in 2.3 to find domain model concepts covering a learner query in a natural language.

In addition to existing modules, the following two will be implemented and integrated:

- the *ULLG Indexer* that works in background and is purposed to maintain the concept mapping matrix (defined in 4.1), the user-to-user similarity matrix and the utility matrix for concepts (defined in 4.2) as well as the ULLG utility matrix (defined in 4.3).
- the *ULLG Recommender* that, given data structures maintained by the ULLG Indexer, choses the best ULLGs to recommend to the current learner.

Moreover some improvements to already existing modules will be done. In particular the *Domain Concepts Selector* will be improved by allowing learners to share to other learners the self-made ULLGs. Such ULLG will be composed by adding selected concepts to a textual description generated from the learner query (optionally enriched by the learner himself). Once shared, the generated ULLG will be accessible to other users.

An additional function that will be integrated in the *ULLG Selector* will be the social rating of ULLGs. Learners will be able to rate ULLGs created by teachers or by learners to provide guidance to other users. This rating will be not exploited by recommender algorithms (that are based on the content of ULLGs rather then on opinions of other users) but can guide the selection process.

The figure 5 shows existing modules (in gray) and modules to be developed (in black) in the context of the ULLG Manager that is part of the IWT application services. The next subsection will provide additional details about the ULLG selection and creation processes from the learner point of view.

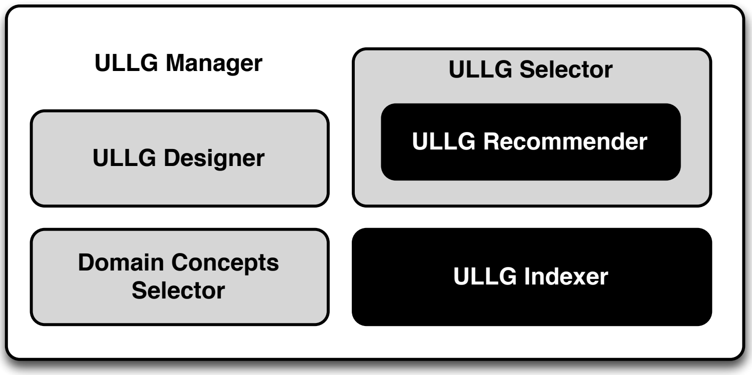

Figure 5. Additional IWT components foreseen.

5.2 ULLG Selection and Creation

Once a learner accesses his *Personal Learning Goals* section he sees a panel composed by three sections (see figure 6). The first section is titled **My Learning Goals**. Here a learner can view and manage his/her ULLGs and study connected courses. By pressing the *access* link he can access the personalized course connected with each learning goal. By pressing the *remove* link he can unsubscribe the course connected with the ULLG. Finally he can rate the learning goal through the classical 5-star mechanism.

Sometimes can happen that icons representing subscribed learning goals exceed the space available in the panel section. In such case 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 through the algorithms defined in section 4. 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 personalized course is subscribed. Under each ULLG, the learner can view the average rating provided by other people that have already used it.

Sometimes can happen that the system has many suggestions and that icons representing suggested goals exceed the space available in the panel section. In such case 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.

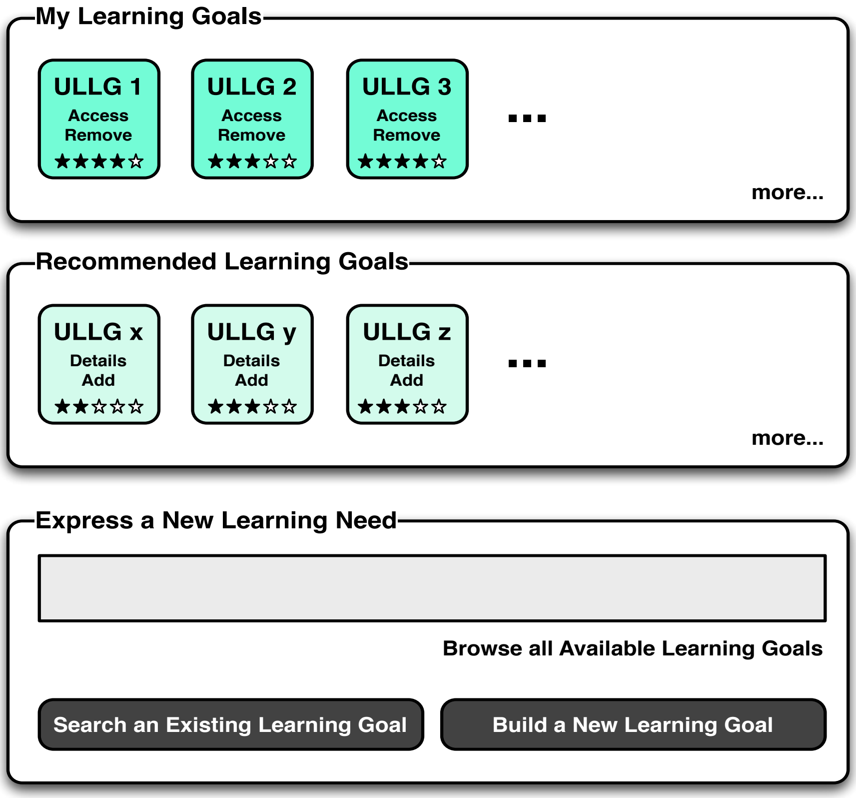

Figure 6. Mock-up of the Personal Learning Goals section.

In the third section titled **Express a New Learning Need** the learner can express a need in a natural language and let the system find a list of suitable ULLGs through the algorithms described in section 2.2 by pressing the button *Search an Existing Learning Goal*. The learner can also build a new learning goal based on the written text through the algorithms described in section 2.3 by pressing the button *Build a New Learning Goal*.

- In the first case the list of compliant ULLGs is presented in a new page where searching and filtering facilities are provided. Here the user can obtain more details of each ULLG, can add it to the list of subscribed ULLGs and can see its average rating. If unsatisfied by the selection, the user can jump to the section described below.
- In the second case, the list of compliant ontology concepts of available domain models is presented to the learner that can add or remove some of them to the new ULLG. Then the learner can add a description (the default description is the one provided in the search text), he can decide to share or not the ULLG and save it. The new ULLG is added to the *My Learning Goals* section with a different icon.

The learner can also decide to browse the list of all available ULLGs by pressing the *Browse Available Learning Goals* link. Also in this case the list is presented in a page where searching and filtering facilities are provided.

6 Related Work

This research falls in the *Recommender Systems* (RS) field, a research field that belongs to *Information Filtering* and is purposed to recommend information items that are likely to be of interest to the user. An introduction to RS has been already provided in section 2 where several kinds of approaches (cognitive, collaborative, hybrid) are presented and, for each of them, several techniques and algorithms are introduced. In the following paragraph we present instead some example of existing recommender system.

Section 6.2 dissertates on the utility of recommendations in Technology Enhanced Learning (TEL) and which are possible roles played by a RS in a TEL environment. Then, section 6.3 presents some example of existing recommender systems for TEL while section 6.4 compares our approach with the one proposed by similar systems. Section 6.5 closes the chapter by introducing some evaluation techniques for TEL recommender systems.

6.1 General Purpose Recommender Systems

Collaborative approaches have always been popular to generate recommendations so that, initially, the RS were called *collaborative filtering systems*. Among the first implementations we include **Tapestry** [13], that, born as an application for e-mail filtering, was able to handle any kind of document, but was designed for relatively small and compact communities.

GroupLens [14] is another collaborative filtering system that was initially applied to Usenet newsgroups. The users could judge each article and an algorithm was used to measure the correlation coefficient and find the degree of agreement between two users. The idea behind GroupLens is currently applied by *MovieLens.com*, a system for recommending movies.

The **IRA** (Intelligent Recommendation Algorithm) [15] system tried to solve the RS *cold start* problem by allowing the propagation of recommendations from one user to another, even if the two user had not even considered a common item. This technique uses a directed graph, where nodes are users and edges represent predictability. Recommendations are calculated using the weighted average of the shortest paths that combine multiple users.

One of the most popular recommender systems is the one used by the **Amazon.com**, website described in [16]. The authors criticize *user-to-user* recommendation techniques since they do not offer scalability and suffer of the *new user* problem. To solve these problems they suggest, for the first time, an *item-to-item* approach faster than the previous one because it depends only on the number of purchases of each user and has good performances even if the user does not have purchased many products.

An example of cognitive algorithm is used for the system **NewsDude** [17] whose aim was to recommend news in a radio broadcast. The user could stop the transmission and send explicit feedback (text or voice) or implicit (if a user was listening to a news for a while, he probably found it interesting). After collecting a number of data, the system used a learning technique to calculate a sequence of news sorted according to the user interests.

The system **Entree** [18] is a prompter of restaurants that uses *case-based reasoning* techniques to select and order some restaurants in U.S. cities. A user could select a restaurant, that already knew, and ask them to look for similar spaces. The system used to describe the restaurant starting to suggest others, placed in order of similarity. The system could also browse other suggested local changing some features according to specific taste so refining their search criteria.

Entree was based on the **FindMe** [19] recommendations technique, which uses examples to guide the search and allows the user to interact with the results, altering the characteristics of the starting example. The FindMe algorithm consists of two parts: a similarity-based analysis, in which the system searches some products similar to those that the user has selected, and a refinement phase, in which products that do not meet the users demand are removed from the results. This approach differs from the simple relevance feedback since the user is aware of the characteristics that influence the filtering process.

Among the hybrid recommendation systems we include **Fab** [6], a system for Web pages filtering that uses a cognitive approach to manage user profiles that, in a second step, are processed with collaborative techniques to find similarities. In [20] it was also proposed a prototype system for suggesting songs that seeks to offer recommendations without relying exclusively on information about previous purchases.

The same authors of the aforementioned Entree have proposed **EntreeC,** a hybrid version [18] which incorporates cascading collaborative techniques for refining the search of the premises. Initially, the functioning EntreeC is similar to that of Entree and relies mainly on the knowledge domain. Unlike Entree, however, EntreeC uses the refining made earlier by others in order to deduce what is the best type of restaurant that the user is looking for.

6.2 Recommendations in Technology Enhanced Learning

Recommendations in a TEL context have many particularities that are based on the richness of the pedagogical theories and models. Differently from buying products, learning is an effort that often takes more time and interactions compared to a commercial transaction. In fact, learners rarely achieve a final end state after a fixed time. Instead of buying a product and then owning it, learners achieve different levels of competences that have various levels in different domains. So, what is important is identifying the relevant learning goals and supporting learners in achieving them.

To build a useful recommender it is important to understand the goals and the tasks for which it is being used within a particular application context. For example in the TEL context, relevant tasks can be supporting learners to achieve a specific learning goal like “providing annotation in context” or “recommending a sequence of learning resources”. The table 9 includes the list of user tasks with a number of specific recommendation goals for TEL that have been related by Herlocker et al. [25]. In particular, given a recommendation task, the table compares goals for a generic recommender and goals for a TEL recommender.

Tasks	Description	Generic recommender	TEL recommenders	New requirements
ANNOTATION IN CONTEXT	Recommendations while user carries out other tasks	E.g. predicting how relevant the links are within a web page	E.g. predicting relevance or usefulness of items in the reading list of a course	Explore attributes for representing relevance or usefulness in a learning context
FIND GOOD ITEMS	Recommendations of suggested items	E.g. receiving list of Web pages to visit	E.g. receiving a selected list of online educational resources around a topic	None
FIND ALL GOOD ITEMS	Recommendation of all relevant items	E.g. receiving a complete list of references on a topic	E.g. suggesting a complete list of scientific literature or blog postings around a topic	None
RECOMMEND SEQUENCE	Recommendation of a sequence of items	E.g. receive a proposed sequence of songs	E.g. receiving a proposed sequence through resources to achieve a particular learning goal	Explore formal and informal attributes for representing relevancy to a particular learning goal
JUST BROWSING	Recommendations out of the box while user is browsing	E.g. people that bought this, have also bought that	E.g. receiving recommendations for new courses on the university site	Explore formal and informal attributes for representing relevance/usefulnes s in a learning context
FIND CREDIBLE RECOMMENDER	Recommendations during initial exploration/testing phase of a system	E.g. movies that you will definitely like	E.g. restricting course recommendations to ones with high confidence or credibility	Explore criteria for measuring confidence and credibility in formal and informal learning

Table 9. Existing user tasks supported by recommender systems.

In comparison to the typical item recommendation scenario, there are several particularities to be considered regarding what kind of learning is desired, e.g. learning a new concept or reinforce existing knowledge may require different type of learning resources. To highlight this aspect, Table 10 shows examples of user tasks that are particularly interesting for TEL.

Tasks	Description	Generic recommender	TEL recommenders	New requirements
FIND NOVEL RESOURCES	Recommendations of particularly new or novel items	E.g. receiving recommendation about latest additions or particularly controversial items	E.g. receiving very new and/or controversial resources on covered topics	Explore recommendation techniques that select items beyond their similarity
FIND PEERS	Recommendations of other people with relevant interests	E.g. being suggested profiles of users with similar interests	E.g. being suggested peer students in the same class	Explore attributes for measuring the similarity with other people
FIND ALL GOOD PATHWAYS	Recommendation of alternative learning paths through learning resources	E.g. receive alternative sequences of similar songs	E.g. receiving a list of alternative learning paths over the same resources to achieve a specific learning goal	Explore criteria for the construction and suggestion of alternative (but similar) sequences

Table 10. User tasks that could be supported by recommender systems.

Thus, although the previously identified user tasks and recommendation goals can be considered valid in a TEL context, there are several particularities and complexities. This means that simply transferring a recommender system from an existing (e.g. commercial) content to TEL may not accurately meet the needs of the targeted users.

In TEL, careful analysis of the targeted users and their supported tasks should be carried out, before a recommendation goal is defined and a recommender system is deployed. So the TEL recommendation goals can be considered rather complex. For this reason a number of context variables have to be considered, such as user attributes, domain characteristics, and intelligent methods that can be engaged to provide personalized recommendations.

In summary, the main aim is the development of a recommendation strategy based on the most relevant information about the individual learner and the available learning activities, historical information about similar learners and activities, guided by educational rules and learning strategies and aimed at the acquisition of learning goals.

Below we assess the existing techniques for recommender systems regarding their applicability and usefulness in TEL. Table 11 provides an initial overview of advantages and disadvantages of each of these approaches and reports the envisaged usefulness of each technique for TEL recommenders [26].

Name	Short description	Advantages	Disadvantages	Usefulness for TEL
Collaborative Filtering (CF) techniques				
User-based CF	Users that rated the same item similarly probably have the same taste. Based on this assumption, this technique recommends the unseen items already rated by similar users.	No content analysis Domain-independent Quality improves Bottom-up approach Serendipity	New user problem New item problem Popular taste Scalability Sparsity Cold start problem	Benefits from experience Allocate learners to groups (based on similar ratings)
Item-based CF	Focus on items, assuming that the items rated similarly are probably similar. It recommends items with the highest correlation (based on ratings for the items).	No content analysis Domain-independent Quality improves over time Bottom-up approach Serendipity	New item problem Popular taste Sparsity Cold start problem	Benefits from experience
Stereotypes or demographics CF	Users with similar attributes are matched, then it recommends items that are preferred by similar users (based on user data instead of ratings).	No cold-start problem Domain-independent Serendipity	Obtaining information Insufficient information Only popular taste Obtaining metadata information Maintenance ontology	Allocate learners to groups Benefits from experience Recommendation from the beginning of the RS
Content-Based (CB) techniques				
Case-based reasoning	Assumes that if a user likes a certain item, s/he will probably also like similar items. Recommends new but similar items.	No content analysis Domain-independent Quality improves over time	New user problem Overspecialisation Sparsity Cold start problem	Keeps learner informed about learning goal Useful for hybrid RS
Attribute-based techniques	Recommends items based on the matching of their attributes to the user profile. Attributes could be weighted for their importance to the user.	No cold-start problem No new user/new item problem Sensitive to changes of preferences Can include non-item-related features Can map from user needs to items	Does not learn Only works with categories Ontology modeling and maintenance is required Overspecialisation	Useful for hybrid RS Recommendation from the beginning

Table 11. Collaborative and content based recommendations in TEL

By analysing the results from the table we can conclude that user and item-based techniques are useful for learning networks which are dealing with different topics (domains). CF techniques can identify high-quality learning activities and enable learners to benefit from the experiences of other successful learners. The bottom-up rating mechanism holds promise for self-directed Learning Networks because no top-down maintenance for identifying high-quality learning activities is required.

CF techniques can be based on pedagogic rules that are part of the recommendation strategy. The characteristics of the current learner could be taken into account to allocate the learners into groups (e.g., based on similar ratings) and to identify the most suitable learning activities. The prior knowledge level of the current learner would then be taken into account to identify the most suitable learning activity.

The stereotype recommendation technique is an accurate way to allocate the learners into groups if no behaviour data is available. In combination with techniques that suffer from the 'cold start' problem, stereotypes complement a recommendation strategy, enabling valuable recommendations from the very beginning.

Case-based reasoning is useful to keep the learner informed about the aimed learning goals. Learning activities which are similar to the ones preferred in the past are recommended to a learner. When a learner wants to reach a higher competence level for the learning goal, the Personal Recommender Systems can also structure the available learning activities by applying pedagogic rules, as defined in the recommendation strategy. This technique complements the recommendation strategy by adding an additional data source for the available learning activities and learners;

The attribute-based techniques can directly map the characteristics of lifelong learners (like the learning goal, prior knowledge, the available study time) to the characteristics of the learning activities. There are learning technology specifications, such as IMS Learning Design that can support this technique through predefined attributes.

6.3 Recommender Systems for TEL

In the TEL domain a number of recommender systems have been introduced in order to propose learning resources to users. Such systems could potentially play an important educational role, considering the variety of learning resources that are published online [27][28][29]. In the following, some recent approaches are reviewed and an assessment of their status of development and evaluation is provided.

One of the first collaborative filtering systems for learning resources has been the **Altered Vista system** [27][28][30]. Its goal was to explore how to collect user-provided evaluations of learning resources, and to propagate them in the form of word-of-mouth recommendations about the qualities of the resources. The team working on Altered Vista used a Collaborative Filtering (CF) technique to explore how the feedback provided by the learners on learning resources can be stored and given back to a community.

Similar research projects in the area of recommending learning resources to learners based on different kind of collaborative filtering techniques is the **RACOFI** (Rule-Applying Collaborative Filtering) Composer System [31][32][33]. The RACOFI methodology is based on the combination of two recommendation approaches by integrating 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 and using them for recommendation. The RACOFI technology is supporting the commercial site **inDiscover** [34] for music tracks recommendation.

The **QSIA** (Questions Sharing and Interactive Assignments) for learning resources sharing, assessing and recommendation has been developed by Rafaeli et al. [35][36]. This system is used in the context of online communities, in order to harness the social perspective in learning and to promote collaboration, online recommendation, and the formation of learner communities. Instead of developing a typical automated recommender system, Rafaeli et al. chose to base QSIA on a mostly user-controlled recommendation process. That is, the user can decide whether to assume control on who advises (friends) or to use a collaborative filtering service. The system has been implemented and used in the context of several learning situations, such as knowledge sharing among faculties and teaching assistants, high school teachers and among students.

The **CYCLADES** system [37] is an interesting step towards a general recommendation service. It also 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. The advantage of the system is the possibility of offering recommendations for learning activities that are developed by different institutions. This approach is currently exemplary for the Open Education Resources movement.

A related system is the **CoFind** prototype [38][39]. It used digital resources that are freely available on the Web but it followed a new approach by applying for the first time folksonomies (tags) 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 typical, neighborhood-based set of collaborative filtering algorithms have been tried in order to support learning object recommendation by **Manouselis et al**. [40]. The innovative aspect of this study is that the engaged algorithms have been multiattribute ones, allowing the recommendation service to consider multi-dimensional ratings that users provide on learning resources.

A different approach to learning resources' recommendation has been followed by **Shen and Shen** [41]. They have developed 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. The rules are fired when gaps in the competencies of the learners are identified, and then appropriate resources are proposed to the learners. A pilot study with the students of a

Network Education college has taken place, providing feedback regarding the users' opinion about the system.

A similar sequencing system has been introduced by Huang et al. [42]. The proposed system, the **Learning Sequence Recommendation System** (**LSRS**), analyzes group-learning experiences to predict and provide a personal list for each learner by tracking others' learning patterns regarding certain topics. This provides learners opportunities to improve their transfer of learning. For example, some learners have studied the course "Management Information System", and then moved on to enroll the course "Data Structure". It is clear that both courses are in different domains. Since both courses are not closely correlated in terms of course continuity, it's difficult to achieve the integration in learning and the transfer of learning. So far as this problem is concerned, LSRS provides a relationship, which is represented as the same concept across the two different domain subjects. The goal is to propose a novel learning mechanism by using the Markov chain model to calculate transition probabilities of possible learning objects in a sequenced course of study.

Tang and McCalla proposed an evolving e-learning system, which includes a hybrid recommendation service [43][44][45][46][47]. Their system is mainly 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 by engaging a Clustering Module (using data clustering techniques to group learners with similar interests) and a Collaborative Filtering Module (using collaborative filtering techniques to identify learners with similar interests in each cluster).

A rather simple recommender system without taking into account any preferences or profile information of the learners was applied by **Janssen et al**. [48]. However, they conducted a large experiment with a control group and an experimental group. They found positive effects on the effectiveness (completion rates of learning objects) though not on efficiency (time taken to complete the learning resources) for the experimental group as compared to the control group.

Nadolski et al. [49] created a simulation environment for different combination of recommendation algorithms in hybrid recommender system in order to compare them against each other regarding their impact on learners in informal learning networks. They compared various cost intensive ontology based recommendation strategies with light-weight collaborative filtering strategies. They concluded that the light-weight collaborative filtering recommendation strategies are not as accurate as the ontology-based strategies but worth-while for informal learning networks when considering the environmental conditions like the lack of maintenance in learning networks. Also, their study reveals that a light-weight collaborative filtering recommendation technique including a rating mechanism is a good alternative to maintain intensive top-down ontology recommendation techniques.

The Mash-Up Personal Learning Environment called **ReMashed** [50][51] recommends learning resources from emerging information of a Learning Network. In ReMashed learners can specify certain Web 2.0 services like Flickr, delicious.com or Sildeshare.com and combine them in a Mash-Up Personal Learning Environment. Learners can rate information from an emerging amount of Web 2.0 information of a Learning Network and train a recommender system for their particular needs. Therefore, ReMashed has three objectives:

1. to provide a recommender system for Mash-up Personal Learning Environments to learners;
2. to offer an environment for testing new recommendation approaches and methods for researchers;
3. to create informal user-generated content data sets that are needed to evaluate new recommendation algorithms for learners in informal Learning Networks.

A hybrid recommendation approach has been adopted in the **CourseRank system** [52] that is used as an unofficial course guide for Stanford University students. In this system, the recommendation process is viewed under the prism of querying a relational database with course and student information [53].

A hybrid approach is also adopted by the **RPL** prototype system that has been implemented in the course repository of the Virtual University of Tunis [54]. 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. The usage logs of the RPL platform are used for this purpose, and a preliminary evaluation experiment has already taken place [55].

Finally, there have been some recent proposals for systems or algorithms that could be used to support recommendation of learning resources. These include a variety of systems, such as a case-based reasoning recommender proposed by Gomez-Albarran and Jimenez-Diaz [56], contextual recommendations that the knowledge-sharing environment of the APOSDLE EU-project [57] offers to the employees of large organizations [58], the A2M prototype [59], recommendation of multimedia learning resources through mobile devices such as cell phones and PDAs have been explored in [60].

6.4 Comparison with Similar Systems

As seen in the previous section, although the research in recommender systems for TEL is very active, many systems still remain research prototypes that have been never used for real applications. Moreover, the few available full systems are currently used inside custom learning applications and none of them is offered as stand-alone product.

The table 12 compares a selection of the available systems and prototypes together and with respect to the prototype tools we are developing and integrating in the ALICE system on the basis of models and algorithms defined in this report.

As it can be seen, the greatest part of available systems uses a classical collaborative approach to recommendation and only few of them hybridize such approach with a more sophisticated one. In such sense our prototype is the only one that relies on ontologies (already present in IWT) to provide better and more fine grained recommendations.

System	Status	Recommendation Approach	Repositories	Sequencing Capabilities	Based on	
					User Knowledge	User Ratings
Altered Vista	Full System	Collaborative	Single			YES
QSIA	Full System	Collaborative	Single			YES
CYCLADES	Full System	Collaborative	Multiple			YES
ELS	Full System	Collaborative + Clustering	Single			YES
CourseRank	Full System	DB Filters	Multiple			
RACOFI	Prototype	Collaborative + Rules Engine	Single			YES
CoFind	Prototype	Collaborative	Web Resources			YES
Shen and Shen	Prototype	Content Based	Single	YES	YES	
LSRS	Prototype	Markov Chains	Single	YES	YES	
Re-Mashed	Prototype	Collaborative	Web 2.0 Resources			YES
RPL	Prototype	Collaborative + Content Based	Single		YES	YES
OUR TOOL	**Prototype**	**Collaborative + Ontology**	**Single**	**YES**	**YES**	**YES**

Table 12. Comparison with similar systems

Our prototype and RPL are the only two prototypes that base recommendations not only on user ratings but also on user knowledge i.e. on concepts that are considered as already known by the learner that is asking for recommendations as well as by other learners.

Moreover, like both the one proposed by Shen and 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 learning resources.

So the proposed tool offers the bigger set of advanced features with respect to competitors. The only limitation is that, given that it is thought to be used together with IWT, it is designed to work on a single rather then on multiple repositories (like a few set of the other systems and prototypes).

6.5 Evaluation of Recommender Systems for TEL

The evaluation of an interactive system ensures that it behaves as expected by the designer and that it meets the requirements of the user [61]. As far as recommender systems in general, and TEL recommenders in particular are concerned, evaluation becomes a critical point at the systems lifecycle for its improvement and success. In fact, until today, evaluation of recommender systems gives emphasis to rather “technical” measures coming from information retrieval research, although the importance of including user-related evaluation methods has been highlighted. In TEL recommender systems evaluation becomes an even more demanding task, considering the particularities of the educational contexts. So, in this section, an overview of relevant evaluation requirements is provided.

In general, evaluating recommender systems and their algorithms is inherently difficult for several reasons. First, different algorithms may be better or worse on different data sets. Second, the goals for which an evaluation is performed may differ. Much early evaluation work focused specifically on the “accuracy” of collaborative filtering algorithms in “predicting” withheld ratings [25]. Recommendation accuracy metrics are classified into three classes.

- *Predictive Accuracy Metrics:* these metrics measure how close the predicted ratings are to true user ratings. Predictive accuracy metrics are particularly important for evaluating tasks in which the predicting rating will be displayed to the user. Examples are *Mean Absolute Error (MAE)*, *Mean Squared Error (MSE)*, *Root Mean Squared Error (RMSE)*, *Normalized Mean Absolute Error (NMAE)*.
- *Classification Accuracy Metrics.* these metrics measure the frequency with which a recommender system makes correct or incorrect decisions about whether an item is good. Example are *Precision*, *Recall*, *F-measure*, *ROC Curves* and related metrics.
- *Rank Accuracy Metrics:* these metrics measure the ability of a recommendation algorithm to produce a recommended ordering of items that matches how the user would have ordered the same items. Examples are *Spearman’s coefficient*, *Kendall’s Tau*, *Half-life Utility Metric*.

There is an emerging understanding that good recommendation accuracy alone does not give users of recommender systems an effective and satisfying experience. In other words recommender systems must provide not just accuracy, but also usefulness. Such measures represent the suitability of the recommendations to users and measure the system utility based on user satisfaction and system performance. Some of these measures are described below.

- *Coverage*: it is the measure of items (item coverage) or users (user coverage) percentage over which the system can form predictions or make recommendations.
- *Confidence*: it indicates how the system is safe for recommendations’ accuracy.
- *Diversity*: it measures the system's ability to make recommendations different to each other.

- *Learning rate*: it is a measure of how fast an algorithm is able to provide "good" recommendations.
- *Robustness*: it represents the stability of the recommendation in the presence of false information. It can also indicate the system stability under extreme conditions.
- *Novelty*: it measures the system's ability to recommend items that the user does not know. It is a necessary condition for the serendipity.
- *Serendipity:* it indicates the ability of the system to make known to user interesting things that he could not have found otherwise.
- *Adaptivity*: It is a measure of the system's ability to adapt to trends and interests of users. It can also indicate the adaptation rate of the system to specific preferences of the user, or to changes in his profile.
- *Scalability*: it indicates how much a recommendation system is scalable to large data sets.
- *Utility*: it represents the gain of the user following the system recommendation.
- *User satisfaction*: it indicates how much the system meets user expectations. It is a parameter difficult to quantify, and dependent on previous metrics.

These general purpose metrics and measures are also useful to evaluate recommender systems in TEL domain. On the other side, by focusing only on technical measures for recommender systems in TEL, without considering the actual needs and characteristics of the learners, is questionable. So, further evaluation procedures that complement the technical evaluation approaches are needed. Common measures to evaluate the success of such systems in educational settings include the following.

- *Effectiveness*: the total amount of completed, visited or studied content objects during a learning phase.
- *Efficiency:* the time that learners need to reach their learning goal.
- *Satisfaction:* the individual satisfaction of the learners with the given recommendations (satisfaction is close to the motivation of a learner and therefore an important measure for learning).
- *Drop-out rate*: the numbers of learners that drop out during the learning phase (in educational research the dropout rate is an important measure when the aim is to graduate as many learners as possible during a learning phase).

Moreover, classical evaluation frameworks from educational research could be adopted and adapted to the recommender systems' context. As an example, the Kirckpatrick's model[62], which measures the success of training using four different layers, could be used to evaluate the success of a recommender system in a TEL context as follows.

- *Reaction of user* i.e. what they thought and felt ("did I enjoy the recommendations I receive?").
- *Learning* i.e. the resulting increase in gaining new knowledge or capabilities ("did I learn what I needed to and get some new ideas, with the help of the recommender?").
- *Behavior* i.e. extent of how acquired knowledge and capability can be implemented/ applied in real life ("will I use the new information and ideas I was recommended?").
- *Results* i.e. effects on the user's performance in the learning or working environment ("do the ideas and information improve my effectiveness and results?").

Therefore, the definition of an overall evaluation framework of TEL recommenders could include the following components.

- A detailed analysis of the evaluation methods and tools that can be employed for evaluating TEL recommendation techniques against a set of criteria proposed for each of the selected components (e.g. user model, domain model, recommendation strategy and algorithm). For the presented example of the Kirckpatrick's dimensions, this would include an identification of the evaluation methods that could be engaged to measure the effect of the recommender in a particular TEL context, upon each one of the four dimensions.
- The specification of evaluation metrics/indicators to measure the success of each component (e.g. evaluating accuracy of the recommendation algorithm, evaluating coverage of the domain model). For the presented example, this would include a specification of the particular metrics that can measure the effect of introducing the recommender in this TEL context.
- The elaboration of a number of methods and instruments that can be engaged in TEL settings to collect evaluation data from engaged stakeholders, explicitly or implicitly, e.g. measuring user satisfaction, assessing impact of the recommender on working tasks, etc. For the presented example, this would include the proposal of specific instruments that can be used to measure each one of the metrics that measure the effect of introducing the recommender in this TEL context.

In summary, the development of concrete evaluation frameworks that will follow a layered approach is an open issue. These frameworks can focus on incorporating as many evaluation dimensions as possible, also addressing pedagogical dimensions, by combining a variety of evaluation methods, metric, and instruments.

In addition, for the various groups of researchers involved in TEL, a number of topics are of high research interest. For example, the recommendation support for learners in formal and informal learning that takes advantage of contextualized recommender systems has become an important one. Also, context awareness could include pedagogical aspects like prior knowledge, learning goals or study time to embed pedagogical reasoning into collaborative filtering driven recommendations.

Another promising approach is the use of multi-criteria input for recommender system in TEL. Users (learners and teachers) can not only rate learning resource based on the level of complexity, curriculum alignment or how much time is required to cover the learning material, but input could also be inferred from different implicit sources. Such multidimensional input can potentially have a high impact on the suitability of recommendations.

7 Conclusions

We defined in this document the theoretical foundations for the management of Upper Level Learning Goals in the ALICE learning system. This document updates and extends [63] (as well as the results presented in the related paper [71]) and takes into account results of interim experimentation activities. With respect to [63]:

- we improved the ULLG recommendation algorithm by adding the calculation of a cognitive component, based on the analysis of existing knowledge structures, for the estimation of the concept utility (this will improve the accuracy of recommendations by also taking into account serendipity);
- we introduced latent factor recommendation models and explained how to use them in the calculation of the collaborative component of the ULLG recommender algorithm to improve performances when the number of users and of concepts increases;
- we reported a detailed example of use of the defined recommendation algorithms in order to demonstrate their effectiveness in a sample case;
- we revised the domain concepts selector software component to allow learners share with other learners the self-made ULLGs created by adding selected concepts to a textual description generated from the learner query;
- we revised the ULLG selector software component in order to let learners rate ULLGs created by teachers or by learners and provide guidance to other users;
- we described in details the ULLG selection and creation processes from the learner point of view in order to guide components' implementation;
- we improved the related work section with a comparison of the prototype resulting from this research with similar systems and research prototypes.

After having developed and integrated with IWT components the defined methodologies, a final experimentation phase will follow. Results coming from that can be used for a further step of methodologies improvement before industrialization.

References

[1] ALICE (Adaptive Learning Via Intuitive/Interactive, Collaborative And Emotional Systems) project, Deliverable D7.1.2 “Models and Methodologies for Knowledge Model Contextualization v2”, project co-funded by the European Commission within the 7th Framework Programme (2007-2013), n. 257639, 2010.

[2] N. Capuano, M. Gaeta, A. Marengo, S. Miranda, F. Orciuoli, P. Ritrovato, LIA: an Intelligent Advisor for e-Learning, Interactive Learning Environments, Taylor & Francis, vol. 17, no. 3, pp. 221-239, September 2009.

[3] Y. Li, D. McLean, Z. Bandar, J. O'Shea, K. A. Crockett, Sentence Similarity Based on Semantic Nets and Corpus Statistics, IEEE TKDE, 18(8), 2006, pp.1138-1150.

[4] Z. Wu, M. Palmer, Verb semantics and lexical selection, Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics, pp. 133-138, 1994.

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

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

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

[8] Perugini, S., Goncalves, M. A., Fox, E. *A. Recommender systems research: A connection-centric survey*. J. Intell. Inf. Syst. 23, 2 (2004), 107–143.

[9] Linden, G., Smith, B., York, J. *Amazon.com recommendations: Item-to-item collaborative filtering.* IEEE Internet Computing 7, 1 (2003), 76–80.

[10] Ungar, L., Foster, D. A *Formal Statistical Approach to Collaborative Filtering*. Proc. of CONALD 98, Conference on Automated Learning and Discovery, Pittsburgh, PA, USA, (2008).

[11] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M. *Combining content-based and collaborative filters in an online newspaper*. Proc. of ACM SIGIR Workshop on Recommender Systems, Berkeley, CA, USA, (1999).

[12] Burke, R. *Hybrid Web recommender systems*. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007).

[13] Goldberg, D., Nichols, D., Oki, B.M., Terry, D. *Using collaborative filtering to weave an information tapestry*. Commun. ACM 35(12) (1992) 61–70.

[14] Resnick, P., N. Iacovou, M. Sushak, P. Bergstrom, J. Riedl. *GroupLens: An Open Architecture for Collaborative Filtering of Netnews*. In Proceedings of 1994 Computer Supported Collaborative Work Conference, ACM Press, (1994) 175–186.

[15] Aggarwal, C. C., J. L. Wolf, K-L. Wu, P. S. Yu. *Horting hatches an egg: A new graphtheoretic approach to collaborative filtering*. In Proc. of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (1999).

[16] Linden, G., B. Smith, J. York. A*mazon.com Recommendations: Item-to-Item Collaborative Filtering*. IEEE Internet Computing, Jan.-Feb. (2003).

[17] Billsus, D., M. Pazzani. *User modeling for adaptive news access*. User Modeling and User-Adapted Interaction, 10, 2-3 (2000) 147-180.

[18] Burke, R. *Knowledge-based Recommender Systems*. In: A. Kent (ed.): Encyclopedia of Library and Information Systems, 69, Sup. 32. (2000).

[19] Burke, R., Hammond, K., and Young, B. The FindMe Approach to Assisted Browsing. IEEE Expert, 12, 4 (1997) 32-40.

[20] Stenzel, R., Kamps, T. *Improving content-based similarity measures by training a collaborative model*. In Proc. Int. Conf. Music Inf. Retrieval – ISMIR (2005) 264–271.

[21] ALICE (Adaptive Learning Via Intuitive/Interactive, Collaborative And Emotional Systems) project, Deliverable D1.1 “Requirements”, project co-funded by the European Commission within the 7th Framework Programme (2007-2013), n. 257639, 2010.

[22] G. Mangione, M. Gaeta, F. Orciuoli, S. Salerno. A Semantic Metacognitive Learning Environment. Cognitive and Metacognitive Educational Systems, AAAI Fall Symposium, 2010.

[23] N. Capuano, M. Gaeta, F. Orciuoli, P. Ritrovato. On-Demand Construction of Personalized Learning Experiences Using Semantic Web and Web 2.0 Techniques. I. Aedo, N. Chen, Kinshuk, D. Sampson, L. Zaitseva (Eds.), Proceedings of the IX IEEE International Conference on Advanced Learning Technologies (ICALT 2009), July 15-17, 2009, Riga, Latvia, pp. 484-488, IEEE Computer Society, 2009.

[24] S. Rendle, L. Schmidt-Thieme. Online-updating regularized kernel matrix factorization models for large-scale recommender systems. Proceedings of RecSys '08, the 2008 ACM conference on Recommender systems, ACM New York, 2008.

[25] Herlocker, J.L. Konstan, J.A., terveen, L.G., & Riedl, J.T.: Evaluating Collaborative Filtering recommender Systems. ACM transactions on Information Systems, 2004

[26] Drachsler H., Hummel H.G. K., Koper R.: Personal recommender systems for learners in lifelong learning: requirements, techniques and model. International Journal of Learning technology, 404-423, 2008

[27] Recker, M.M., Wiley, D.A.: A non-authoritative educational metadata ontology for filtering and recommending learning objects. Interactive learning environments 9(3), 255–271 (2001).

[28] Recker, M.M., Walker, A., Wiley, D.: An interface for collaborative filtering of educational resources. In: International Conference on Artificial Intelligence, pp. 26-29. Las Vegas, Nevada, USA (2000).

[29] Kumar, V., Nesbit, J., Han, K.: Rating Learning Object Quality with Distributed Bayesian Belief Networks: The Why and the How. In: Proc. of the Fifth IEEE International Conference on Advanced Learning Technologies, ICALT’05 (2005).

[30] Recker, M.M., Walker, A., Lawless, K.: What do you recommend? Implementation and analyses of collaborative information filtering of web resources for education. Instructional Science 31(4/5), 299–316 (2003).

[31] Anderson M., Ball M., Boley H., Greene S., Howse N., Lemire D., McGrath S.: RACOFI: A Rule-Applying Collaborative Filtering System. Paper presented at the conference IEEE/WIC COLA 2003, 13 October, Halifax, Canada.

[32] Baudisch P.: Dynamic Information Filtering. PhD Thesis, GMD Forschungszentrum Informationstechnik GmbH, Sankt Augustin (2001).

[33] Lemire, D., Boley, H., McGrath, S., Ball, M. (2005). Collaborative Filtering and Inference Rules for Context-Aware Learning Object Recommendation. International Journal of Interactive Technology and Smart Education 2(3) (2005).

[34] inDiscover, http://www.indiscover.net

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

[36] Rafaeli , S., Dan-Gur , Y., Barak , M.: Social Recommender Systems: Recommendations in Support of E-Learning. International Journal of Distance Education Technologies 3(2), 29–45 (2005).

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

[38] Dron, J., Mitchell, R., Boyne, C., & Siviter, P.: CoFIND: steps towards a self-organising learning environment. Proceedings of the World Conference on the WWW and Internet (WebNet 2000), San Antonio, Texas, USA, October 30-November 4, pp. 146-151, USA AACE (2000a).

[39] Dron, J., Mitchell, R., Siviter, P., & Boyne, C.: CoFIND-an experiment in n-dimensional collaborative filtering. Journal of Network and Computer Applications 23(2), 131–142 (2000b).

[40] Manouselis, N., Vuorikari, R., Van Assche, F.: Simulated Analysis of MAUT Collaborative Filtering for Learning Object Recommendation. In: Proc. of the Workshop on Social Information Retrieval in Technology Enhanced Learning (SIRTEL 2007). Crete, Greece (2007).

[41] Shen, L., Shen, R.: Learning content recommendation service based-on simple sequencing specification. In: Liu W et al. (eds), pp. 363-370. Lecture notes in computer science (2004).

[42] Huang, Y.-M., Huang, T.-C., Wang, K.-T., Hwang, W.-Y.: A Markov-based Recommendation Model for Exploring the Transfer of Learning on the Web. Educational Technology & Society 12(2), 144–162 (2009).

[43] Tang T., McCalla G.: Smart Recommendation for an Evolving E-Learning System. In: Proc. of the Workshop on Technologies for Electronic Documents for Supporting Learning, International Conference on Artificial Intelligence in Education (AIED 2003) (2003).

[44] Tang, T., McCalla, G.: Utilizing Artificial Learner on the Cold-Start Pedagogical-Value based Paper Recommendation. In: Proc. of AH 2004: International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (2004a).

[45] Tang, T., McCalla, G.: Beyond Learners' Interest: Personalized Paper Recommendation Based on Their Pedagogical Features for an e-Learning System. In: Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence (PRICAI 2004), pp. 301-310 (2004b).

[46] Tang, T.Y., McCalla, G.: On the pedagogically guided paper recommendation for an evolving web-based learning system. In: Proceedings of the 17th International FLAIRS Conference, pp. 86-91 (2004c).

[47] Tang, T., McCalla, G.: Smart Recommendation for an Evolving E-Learning System: Architecture and Experiment. International Journal on E-Learning 4(1), 105–129 (2005).

[48] Janssen, J., Tattersall, C., Waterink, W., Van den Berg, B., Van Es, R., Bolman, C., et al.: Self-organising navigational support in lifelong learning: how predecessors can lead the way. Computers & Education 49, 781–793 (2005).

[49] Nadolski, R.J., Van den Berg, B., Berlanga, A., Drachsler, H., Hummel, H., Koper, R., Sloep, P.: Simulating Light-Weight Personalised Recommender Systems in Learning Networks: A Case for Pedagogy-Oriented and Rating-Based Hybrid Recommendation Strategies. Journal of Artificial Societies and Social Simulation (JASSS), 12(14) (2009).

[50] Drachsler, H., Pecceu, D., Arts, T., Hutten, E., Rutledge, L., Van Rosmalen, P., Hummel, H.G.K., Koper, R.: ReMashed-Recommendations for Mash-Up Personal Learning Environments. In: Cress, U., Dimitrova, V., Specht, M. (eds.): Learning in the Synergy of Multiple Disciplines, Proceedings of the 4th European Conference on Technology Enhanced Learning (EC-TEL 2009), LNCS 5794, pp. 788-793, Berlin: Springer (2009a).

[51] Drachsler, H., Pecceu, D., Arts, T., Hutten, E., Rutledge, L., Van Rosmalen, P., Hummel, H.G.K., Koper, R.: ReMashed-An Usability Study of a Recommender System for Mash-Ups for Learning. 1st Workshop on Mashups for Learning at the International Conference on Interactive Computer Aided Learning, Villach, Austria (2009b).

[52] CourseRank system, https://courserank.stanford.edu/CourseRank/main

[53] Koutrika, G., Ikeda, R., Bercovitz, B., Garcia-Molina, H.: Flexible Recommendations over Rich Data. In: Proc. of the 2nd ACM International Conference on Recommender Systems (Rec-Sys'08). Lausanne, Switzerland, (2008).

[54] RPL platform, http://cours.uvt.rnu.tn/rpl

[55] Khribi, M.K., Jemni, M., Nasraoui, O.: Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval. Educational Technology & Society 12(4), 30–42 (2009).

[56] Gomez-Albarran M., Jimenez-Diaz G.: Recommendation and Students'Authoring in Repositories of Learning Objects: A Case-Based Reasoning Approach. International Journal of Emerging Technologies in Learning (iJET) 4(1), 35–40 (2009).

[57] APOSDLE EU-project, http://www.aposdle.tugraz.at

[58] Aehnelt M., Ebert M. Beham G., Lindstaedt S., Paschen A.: A Socio-technical Approach towards Supporting Intra-organizational Collaboration. In: Dillenbourg P. and Sprecht M., (eds.) Times of convergence: Technologies across learning contexts. Proceedings of the 3rd European Conference on Technology Enhanced Learning (EC-TEL 2008), Maastricht, The Netherlands, LNCS 5192, pp. 33-38, Berlin: Springer (2008).

[59] Santos, O.C.: A recommender system to provide adaptive and inclusive standard-based support along the eLearning life cycle. In: Proceedings of the 2008 ACM conference on Recommender systems, pp. 319-322 (2008).

[60] Klamma, R., Spaniol, M., Cao, Y.: Community Aware Content Adaptation for Mobile Technology Enhanced Learning. In: Innovative Approaches for Learning and Knowledge Sharing, pp. 227-241 (2006).

[61] Dix, A. J., Finlay, J. E., Abowd, G. D., and Beale, R.: Human-Computer Interaction. Harlow, England: Prentice Hall (1998).

[62] Kirkpatrick, D.L.: Evaluating Training Programs (2nd ed.). Berrett Koehler, San Francisco (1959).

[63] ALICE (Adaptive Learning Via Intuitive/Interactive, Collaborative And Emotional Systems) project, Deliverable D7.2.1 “Models and Methodologies for Upper Level Learning Goals Support v1”, project co-funded by the European Commission within the 7th Framework Programme (2007-2013), n. 257639, 2010.

[64] Yehuda Koren , Robert Bell , Chris Volinsky, Matrix Factorization Techniques for Recommender Systems, Computer, v.42 n.8, p.30-37, August 2009.

[65] Y. Koren, “Factorization Meets the Neighbourhood: A Multifaceted Collaborative Filtering Model”, Proc. 14th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining, ACM Press, 2008, pp. 426-434.

[66] B.M. Sarwar et al., “Application of Dimensionality Reduction in Recommender System - A Case Study,” Proc. KDD Workshop on Web Mining for e-Commerce: Challenges and Opportunities (WebKDD), ACM Press, 2000.

[67] G. Takacks et al., “Major Components of the Gravity Recommendation System,” SIGKDD Explorations, vol. 9, 2007, pp. 80-84.

[68] R. Bell and Y. Koren, “Scalable Collaborative Filtering with Jointly Derived Neighbourhood Interpolation Weights,” Proc. IEEE Int’l Conf. Data Mining (ICDM 07), IEEE CS Press, 2007, pp. 43-52.

[69] Y. Koren, “Collaborative Filtering with Temporal Dynam- ics,” Proc. 15th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (KDD 09), ACM Press, 2009, pp. 447-455.

[70] S. Rendle, L. Schmidt-Thieme: Online-Updating Regularized Kernel Matrix Factorization Models for Large-Scale Recommender Systems. RecSys 2008.

[71] N. Capuano, M. Gaeta, P. Ritrovato, S. Salerno. A Recommender System for Learning Goals. Accepted at the 4th World Summit on the Knowledge Society (WSKS 2011), September 21-23, 2011, Mykonos, Greece.