An Educational Recommender System based on Argumentation Theory

Paula Rodríguez [a] Stella Heras [b] Javier Palanca [b]
Jhon M. Poveda [a] Néstor Duque [c] Vicente Julián [b]

[a] *Universidad Nacional de Colombia*
Calle 59 A N 63-20, Medellín, Colombia
Email: parodriguezma@unal.edu.co
[b] *Universitat Politècnica de València*
Camino de Vera s/n, 46022 Valencia, Spain
Email: {sheras, jpalanca, vinglada}@dsic.upv.es
[c] *Universidad Nacional de Colombia*
Cra 27 N 64-60 Manizales, Caldas, Colombia
Email: ndduqueme@unal.edu.co

Recommender Systems aim to provide users with search results close to their needs, making predictions of their preferences. In virtual learning environments, Educational Recommender Systems deliver learning objects according to the student's characteristics, preferences and learning needs. A learning object is an educational content unit, which once found and retrieved may assist students in their learning process. In previous work, authors have designed and evaluated several recommendation techniques for delivering the most appropriate learning object for each specific student. Also, they have combined these techniques by using hybridization methods, improving the performance of isolated techniques. However, traditional hybridization methods fail when the learning objects delivered by each recommendation technique are very different from those selected by the other techniques (there is no agreement about the best learning object to recommend). In this paper, we present a new recommendation method based on argumentation theory that is able to combine content-based, collaborative and knowledge-based recommendation techniques, or to act as a new recommendation technique. This method provides the students with those objects for which the system is able to generate more arguments to justify their suitability. It has been implemented and tested in the Federation of Learning Objects Repositories of Colombia, getting promising results.

Keywords: Educational Recommender Systems, Argumentation

1. Introduction

According to the IEEE, a learning object (LO) can be defined as a digital entity involving educational design characteristics. Each LO can be used, reused or referenced during computer-supported learning processes, aiming at generating knowledge and competences based on student's needs [1]. LOs have functional requirements such as accessibility, reuse, and interoperability. The concept of LO requires understanding of how people learn, since this issue directly affects the LO design in each of its three dimensions: pedagogical, didactic, and technological [2]. In addition, LOs have metadata that describe and identify the educational resources involved and facilitate their searching and retrieval. Learning Objects Repositories (LORs), composed of thousands of LOs, can be defined as specialized digital libraries storing several types of heterogeneous resources. LORs are currently being used in various e-learning environments and belong mainly to educational institutions [2,3]. Also, federations of LORs provide educational applications to search, retrieve and access specific LO contents available in any LOR [4].

Recommender Systems aim to provide users with search results close to their needs, making predictions of their preferences. In virtual learning environments, Educational Recommender Systems (ERS) deliver LOs according to the student's characteristics, preferences and learning needs [5]. In order to improve recommendations, ERS must perform feedback processes and implement mechanisms that enable them to obtain a large amount of information about users and how they use the LOs. ERS can be classified into several types [6]:

- Content-based ERS: in this kind of systems, recommendations are performed based on the user's profile and created from the content analysis of the LOs that the user has already assessed in the past. The content-based systems use "item-by-item" algorithms generated through the association of correlation rules among those items.

AI Communications
ISSN 0921-7126, IOS Press.

- Collaborative ERS: these systems hold great promise for education, not only for their purposes of helping learners and educators to find useful educational resources, but also as a means of bringing together people with similar interests and beliefs, and possibly as an aid to the learning process itself. In this case, the recommendations are based on a similarity degree among users. Collaborative filtering algorithms aim at suggesting new items or predicting the utility of a certain item for a particular user profile based on the choices of other similar user profiles.
- Knowledge-based ERS: these systems attempt to suggest LOs based on inferences about the user's needs and preferences. Knowledge-based approaches use knowledge about how a particular item meets a particular user need, and can therefore reason about the relationship between a need and a possible recommendation. In addition, these systems are based on the user's browsing history and his/her previously selected LOs.
- Hybrid Recommender Systems: the hybrid approach combines several ERS techniques in order to maximise the advantages of each one and, thus, make better recommendations. To make the hybridization of recommendation techniques –using at least two of them– Burke [6] describes different methods that could be applied (e.g. weighted, switching, mixed, cascade, feature combination, feature augmentation, and meta-level).

Recommending LOs presents some challenges that need to be addressed in order to design a recommender system suitable for e-learning environments [7]. Some of these challenges are that each learner uses her own learning process, based on her own tools, methods and paths, that the recommender system needs to take into account.

The rest of the paper is structured as follows: section 2 presents the motivation of this work, section 3 reviews related work, section 4 presents our argumentation-based hybrid recommendation method, in section 5 we provide an example of how our system works, in section 6 we evaluate our proposal, and finally, section 7 presents conclusions and future work.

2. Motivation

In previous work, authors have proposed a Student-Centered Hybrid ERS, designing and evaluating several recommendation techniques for delivering the most appropriate LO for each specific student [8,9]. Also, they have combined these techniques by using hybridization methods, improving the performance of isolated techniques. The ERS proposed follows a hybrid recommendation technique that combines content-based, collaborative and knowledge-based approaches. In the system, LOs are retrieved from LORs and federations of LORs, using the stored descriptive *metadata* for these objects. Concretely, our ERS follows the *IEEE-LOM*[1] standard to represent the metadata about the LOs. This is a hierarchical data model that defines around 50 metadata fields clustered into 9 categories. Figure 1 shows the fields used in our ERS (highlighted in bold). Also, *student profiles*, including their personal information, language, topic and LO's format preferences, educational level, and learning style (auditory, kinaesthetic, reader, or visual), are used by the system to generate recommendations.

Therefore, as shown in Figure 2, the ERS is composed by six modules: three recommendation modules (one for each recommendation technique: content-based, collaborative and knowledge-based); a module that performs the hybridization (integration) process, to provide the student with the most relevant and appropriate LOs by using a subset of those recommended by each recommendation module[2]; and, finally, two modules that handle information about student profiles and LOs metadata.

Figure 3 shows the specific LOs metadata and students' profile data that each recommendation modules uses.

The motivation of this paper started when we detect several disadvantages in this hybridization process. On the one hand, it does not take the relevance of the LOs into account to encourage the use of a specific LO over another (considering that a LO is relevant for a student if it matches his/her learning objectives and profile). On the other hand, it fails when the LOs delivered by each recommendation technique are very different from those selected by the other techniques (there is no agreement about the best LO to recommend).

In this work we first analysed the incidence of this problem. We performed some experiments to determine the dispersion degree between the LOs proposed

[1] 1484.12.1-2002 - Institute of Electrical and Electronics Engineers (IEEE) Standard for Learning Object Metadata: `https://standards.ieee.org/findstds/standard/1484.12.1-2002.html`

[2] Several hybridization methods, as proposed in [6], were tested in [9].

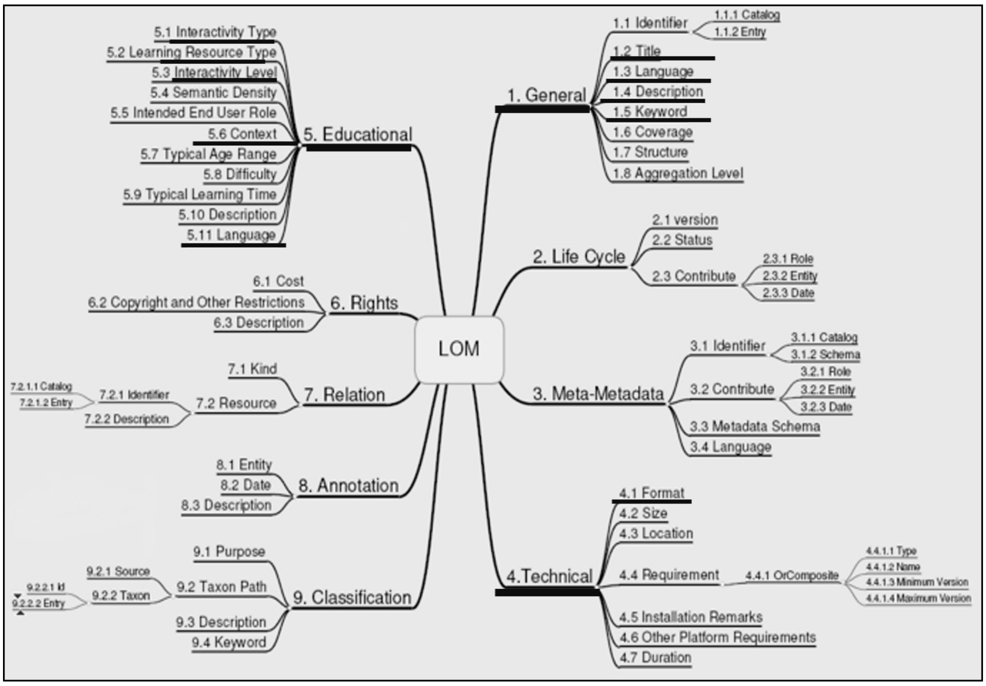

Fig. 1. IEEE-LOM metadata used in the ERS.

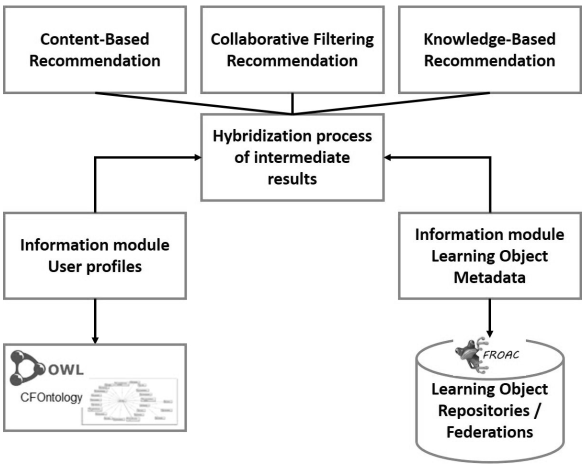

Fig. 2. Student-Centered Hybrid ERS.

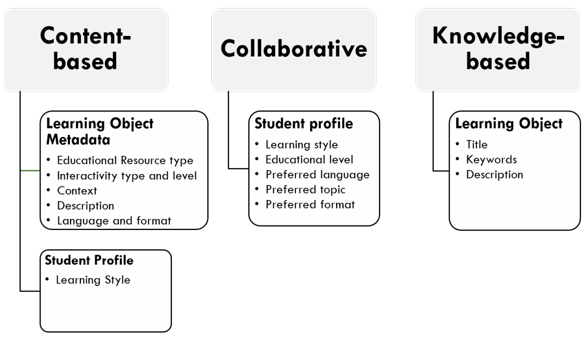

Fig. 3. LOs metadata and students' profile data used by the recommendation modules.

by each recommendation technique (to determine how different are the top 5 or the top 10 LOs proposed by the three recommendation modules). Dispersion tests were performed as follows:

1. A student with a visual learning style was selected.
2. A search on the federation of repositories was performed to retrieve LOs about the topic (keyword) *Algorithms*.
3. The top 5 and top 10 results provided by each recommendation module (content, collaborative and knowledge-based) were saved for analysis.
4. The process was repeated with other keywords (*Programming* and *Audit*).
5. The process was repeated with other students with auditory and kinaesthetic learning styles.

Then, the amount of LOs that overlap between the three recommenders for each iteration of the tests was computed. The average of the results are shown in Figure 4.

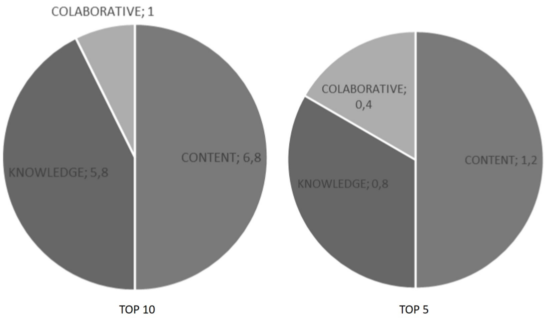

Fig. 4. Results of the dispersion tests.

Results show that, in many cases, the traditional hybridization methods widely disagree in the best LOs to recommend (there is no intersection between the recommendations provided by the three recommendation modules) and hence, this negatively affects their performance. For instance, knowledge-based recommendations on the Top 5 tests, result in an average dispersion of 0.8 LOs. This means that for each 5 LOs delivered by this recommendation module, on average, only 0.8 overlap with the results of the other techniques (there is no agreement among them).

Finally, to overcome this problem, in this paper we also present a new recommendation method based on argumentation theory. Among the wide range of agreement technologies proposed in the last years [10], argumentation provides a natural means of dealing with conflicts and knowledge inconsistencies with a high resemblance with the way in which human societies reach agreements [11]. Our method is able to combine content-based, collaborative and knowledge-based recommendation techniques, or to act as a new recommendation technique, providing students with those LOs for which the system is able to generate more arguments to justify their suitability.

3. Related work

Over the last years, the literature on ERS reports a growing interest in the area [7]. In [12], authors discuss the need of support tools for learners based on contextualised recommender systems. According to the authors, it is very important to take into account pedagogical aspects, like prior knowledge, learning goals or study time in the recommendation process. In addition, they argue that the development of concrete evaluation frameworks that follow a layered approach is still an open research issue. These frameworks may focus on incorporating as many evaluation dimensions as possible, on addressing pedagogical dimensions, or on combining a variety of evaluation methods, metrics, and instruments.

In this regard, in [13] a recommendation system based on genetic algorithms that performs two recommendation processes was proposed. The first one uses explicit characteristics represented in a matrix of student's preferences, while the second assigns implicit weights to educational resources that are considered as chromosomes in a genetic algorithm that optimises them by using historical values. However, compared with out proposal, this work does not perform hybrid recommendation, but combines the characteristics of the student profile. Following a hybrid approach similar to ours, Zapata et al. deliver educational materials adapted to the user profile by combining several types of filtering methods with the available information about LOs and users [14]. However, although this work combines several filtering criteria (content-based, collaborative activity, and demographics), it is aimed at helping teachers rather than students. By contrast, the research presented by Sikka et al. [3], which presents an e-learning environment to recommend learning materials by using web mining techniques and software agents, implements just a unique collaborative recommendation filter rather than using a hybrid approach. However, in [15] a review of some hybrid recommendation systems was performed, concluding that the hybrid filter obtained by integrating collaborative and content-based filtering approaches improves the predictions made by the recommender. We share this view and extend it to recommend educational materials recovered from LORs. For a current overview on the panorama of recommender systems to support learning, we refer the reader to the work presented in [5].

Traditional recommender systems base their recommendations on quantitative measures of similarity, but fail at using the qualitative data available to empower recommendations [16]. Usually, recommender systems do not provide an explanation about the reasoning process that has been followed to come up with specific recommendations. However, people rely more on recommendations when the system can also show the reasons behind the recommendations [17], and when they can understand the reasons why these recommendations are presented [18]. Moreover, even when users already know the recommendations presented, the latter work demonstrated that they prefer recommender systems that are able to justify their suggestions. Thus, what is understood as a good recommendation is changing from the one that minimises some error evaluation to the one that is really able to *persuade* people and make them happier.

Other issue to address in ERSs is group recommendation. There are different processes and algorithms to recommend a LO to a student or to a group of students, where the learning profile is a unified profile computed from the group preferences [19]. However, in this work we are not focusing on group recommendation, since our goal is to improve recommendations made to single students taking into account their learning profile.

Recently, some argument-based recommender systems and recommendation techniques have been proposed to recommend music [20], news [21], movies [22], or restaurants [23], or to perform content-based web search [24]. Among them, we share the approach of the movie recommender system based on defeasible logic programming proposed in [25]. In this work, authors define a preset preference criteria between rules to resolve argument attacks. However, as will be explained in section 4, we use a probabilistic method to compute the likelihood that an argument prevails over another, which makes the system more adaptive. In educational domains, argumentation theory and tools have a large history of successful applications, specially to teach critical thinking skills in law courses [26]. However, to the best of our knowledge, the application of argumentation theory to enhance ERS is a new area of research.

There are a number of open challenges for the application of argumentation theory to recommender systems [24], such as *exposing underlying assumptions behind recommendations*, *approaching trust and trustworthiness* from the perspective of backing recommendations and *providing rationally compelling arguments for recommendations*. Our work involves a contribution in this latter area.

4. Formal Framework

In this section, we provide an overview on the argumentation formalism used for our proposal. As pointed out in section 2, the original Student-Centered Hybrid ERS proposed uses several sources of knowledge to generate LOs recommendations for the students, namely information about the *student profile* and *metadata* about the LOs to recommend. In this paper, we present a hybrid recommendation method based on argumentation theory that uses these sources of knowledge and provides the students with those LOs for which the system is able to generate more arguments to justify their suitability. Concretely, we use a defeasible argumentation formalism based on logic programming (*DeLP*, see [27] for details) to encode information about the *facts* (metadata and profiles data) and the *rules* that determine the allowed inferences that can be done in our system.

Definition 1 (DeLP) *A defeasible logic program* P = (Π,Δ)*, models strict (*Π*) and defeasible (*Δ*) knowledge about the application domain. In our system, the set* Π *includes strict inference rules with empty body that represent facts. Correspondingly, the set* Δ *includes defeasible rules of the form* $P \leftarrow Q_1,...,Q_k$*, which represent the defeasible inference that literals* $Q_1,...,Q_k$ *may provide reasons to believe P.*

For instance, $auditory(jose)$ represents the fact that a student named *'jose'* has an auditory learning style and prefers materials with sounds, and auditory formats such as mp3, mp4, or avi. Facts are assumed to be non-contradictory (e.g., if $\sim$ represents default logic negation, $auditory(jose)$ and $\sim auditory(jose)$ cannot be inferred). Table 1 shows a compendium of these rules[3]. These rules are divided on 4 groups, 3 to represent the knowledge used by each recommendation technique (content-based, collaborative or knowledge-based), and 1 to represent general domain knowledge. Section 5 provides an example to clarify their meaning and use.

Given a DeLP, the program can be queried to resolve if a ground literal can be derived from the program, and hence supported by an argument(s) based on the rules of Δ. Concretely, for our hybrid recommendation method to recommend a LO to a specific user, we need to be able to derive any of the *recommend(user, LO)* defeasible rules from our DeLP. Furthermore, we have also designed a module for constructing explanations (arguments) based on these rules. Since the number of rules of our ERS is finite and small, currently this is a simple module that associates each rule with a scheme of explanation (see table 2 for an example).

[3]The complete rule set is not provided due to space limitations.

GENERAL RULES	*G1: ∼recommend(user, LO) ← cost(LO) > 0* *G2: ∼recommend(user, LO) ← quality_metric(LO) < 0.7*
CONTENT-BASED RULES	*C1: recommend(user, LO) ← educationally_appropriate(user, LO) ∧ generally_appropriate (LO)*
	C1.1: educationally_appropriate(user, LO) ← appropriate_resource(user, LO) ∧ appropriate_interactivity(user, LO) *C1.1.1: appropriate_resource(user, LO) ← user_type(user, type) ∧ resource_type(LO, type)* *C1.1.2: appropriate_interactivity (user, LO) ← user_type(user, type) ∧ interactivity_type(LO, type)* *C1.2: generally_appropriate(LO) ← structure(LO, atomic) ∧ state(LO, final)*
	C2: recommend(user, LO)← educationally_appropriate(user, LO) ∧ generally_appropriate(LO)) ∧ technically_appropriate(user, LO)
	C2.1: technically_appropriate(user, LO) ← appropriate_language(user, LO) ∧ appropriate_format(LO) *C2.1.1: appropriate_language(user, LO) ← language_preference(user, language) ∧ object_language(LO, language)* *C2.1.2: appropriate_format(LO) ← format_preference(user, format) ∧ object_format(LO, format)*
	C3: recommend(user, LO) ← educationally_appropriate(user, LO) ∧ generally_appropriate (LO) ∧ updated(LO)
	C3.1: updated(LO) ← date(LO, date) < 5 years
	C4: recommend(user, LO) ← educationally_appropriate(user, LO) ∧ generally_appropriate(LO) ∧ learning_time_appropriate(LO)
	C4.1: learning_time_appropriate(LO) ← hours(LO) < γ
COLLABORATIVE RULES	*O1: recommend(user1, LO) ← similarity(user1, user2) > α ∧ vote(user2, LO) ≥ 4*
KNOWLEDGE-BASED RULES	*K1: recommend(user1, LO)← similarity(LO1, LO2) > β ∧ vote(user1, LO2) ≥ 4*

Table 1

Defeasible rules

Rule	Explanation	Description
$C1$	$E1$	The learning object LO fits the topic T, is suitable for your LS learning style, and it is atomic and stable.
$C2$	$E2$	The learning object LO fits the topic T, is suitable for your LS learning style, and fits your L language and F format preferences.
$C3$	$E3$	The learning object LO fits the topic T, is suitable for your LS learning style, fits your L language and F format preferences, and it is updated.
$C4$	$E4$	The learning object LO fits the topic T, is suitable for your LS learning style, and fits your L language, F format preferences and learning time < T preferences.
$O1$	$E5$	The system has found a user that whose profile is similar to yours who liked LO
$K1$	$E6$	The system has found that you liked LOx, which is similar to LOy.

Table 2

Explanation schemes.

Arguments in this framework are defined as follows:

Definition 2 (Argument) *An* argument $\mathcal{A}$ *for h (represented as a pair* $\langle \mathcal{A}, h \rangle$*) is a minimal non-contradictory set of facts and defeasible rules that can be chained to derive the literal (or conclusion) h.*

Then, arguments are generated by backward chaining of both facts and defeasible rules, a mechanism similar to the *Selective Linear Definite (SLD)* derivation of standard logic programming. Therefore, recommendations are computed by chaining arguments in a recursive process that creates a *dialectical tree* (see [27]) whose root node is the original argument under discussion (i.e. whether to recommend or not a LO for a particular user), and whose children nodes are arguments that defeat their parents.

Arguments can be *attacked* by other arguments that *rebut* them (i.e. propose the opposite conclusion) or *undercut* them (i.e. attack clauses of their body).

Definition 3 (Attack) *An argument* $\langle \mathcal{B}, q \rangle$ attacks *another argument* $\langle \mathcal{A}, h \rangle$ *if we can derive* $\sim h$ *from* $\mathcal{B}$ *or if q implies that one of the clauses of* $\mathcal{A}$ *no longer holds (there are a sub-argument* $\langle \mathcal{A}_1, h_1 \rangle$ *from* $\langle \mathcal{A}, h \rangle$ *such that* $\Pi \cup \{h_1, q\}$ *is contradictory).*

Therefore, an argument for not recommending a LO can be generated if an argument for recommending is attacked. Note that we assume negation as failure, so an argument for not recommending a LO can be generated by chaining rules whose literals cannot be derived (we do not have information to resolve them). For instance, by using the rule *O1*, which recommends a LO for a *user1* if other similar *user2* likes that object (i.e. *user2* has voted the LO with a score greater than 4), we can derive an argument for not recommending the LO: 1) if the system cannot find a similar user (negation as failure); or 2) if there is a similar user and he/she does not like the LO (undercut).

To resolve attacks between arguments, each rule has an associated probability measure that estimates the probability that an argument (generated by using the rule) succeeds based on the aggregated probability of the clauses that form the body of the rule. In doing so, we use a simplified *probabilistic argumentation* framework [28] that assigns probability values to arguments and aggregates these probabilities to compute a suitability value to rank and recommend LOs.

Definition 4 (Argumentation Framework) *In our ERS, an argumentation framework is a tuple* (Arg, P_{Arg}, D) *where* Arg *is a set of arguments,* $D \subseteq Arg \times Arg$ *is a defeat relation, and* $P_{Arg} :\rightarrow [0:1]$ *is the probability that an argument holds.*

The probability of an argument $Arg = \langle \mathcal{A}, h \rangle$ is calculated as follows:

$$P_{Arg} = \begin{cases} 1, if \mathcal{A} \subseteq \Pi \\ \dfrac{\sum_{i=1}^{k} P_{Q_i}}{k}, if \mathcal{A} \subseteq \Delta \mid h \leftarrow Q_1, ..., Q_k \end{cases} \quad (1)$$

Facts are assumed to have probability 1. The probability of defeasible rules is computed as the average of the probabilities of the literals $Q_1, ..., Q_k$ that form their body (i.e. 1 if they are facts, 0 if they cannot be resolved, or P_{Q_i} if they are derived from other defeasible rules).

Definition 5 (Defeat) *In our ERS, an argument* $\langle \mathcal{B}, q \rangle$ defeats *another argument* $\langle \mathcal{A}, h \rangle$ *if* $\mathcal{B}$ *attacks* $\mathcal{A}$ *and* $P_B > P_A$.

5. Running Example

Students query our ERS to get LO recommendations that may fit their learning objectives and preferences. With this aim, the system has a search engine that allows a student to find LOs by using keywords that express the educational skills that they want to achieve. This search results in a list of LOs that match the keywords. After that, our ERS starts the recommendations process to rank and deliver LOs of this list: the content-based recommendation module triggers its inference rules by using the LOs metadata and the student's learning style; the collaborative recommendation module seeks similar user profiles to deliver items that have been evaluated by similar students; and the knowledge-based recommendation module determines whether any LO in the list is similar to another LO that the student has already used and assessed positively. Then, the new argumentation-based hybridization method is used to combine these three sets of LOs and deliver those for which the system can generate better arguments to justify their suitability for the search performed by the student.

To illustrate the operation of our method, in this section we show a running example using the LOs stored in the FROAC[4] repository (the Federation of Learning Objects Repositories of Colombia) [2]. FROAC has 637 LOs indexed, stored in different repositories. The main topics of the LOs stored are: Analysis and design of algorithms and information systems, audit, databases, software engineering, artificial intelligence, programming, natural sciences, social sciences, computing, and mathematics. FROAC was developed at the *Universidad Nacional de Colombia*, as a result of a research project entitled *ROAC, Creación de un modelo para la Federación de OA en Colombia que permita su integración a confederaciones internacionales* of *COLCIENCIAS*. FROAC also stores information about its user's profiles (students). For each student, FROAC stores explicit features such as personal information (e.g. full name, date of birth, email, gender, and language), LO preferences (language, topic, and format), and psycho-pedagogical information (learning style). The students' learning style is obtained through a test with 24 questions that determine how the student processes the information that he/she receives and turns it into knowledge. The students of the National University of Colombia make an intensive use of FROAC. However, they have difficulties in specifying a query

[4]FROAC: http://froac.manizales.unal.edu.co

string that meets what they really want to find. Therefore, our ERS was implemented to help those students to find materials to support their learning. Furthermore, students also reported difficulties to understand why the system selects a specific LO over the list of potential candidates as the one that best fits their learning objectives. Thus, we have designed the new argumentation-based hybridization module not only with the objective of improving the quality of recommendations, but also with the aim of being able to provide the students with justifications for those recommendations.

Let us assume that a student with an *auditory* learning style (he prefers auditory LOs with formats such as mp3, mp4, avi, etc.), has queried the ERS to find LOs that can help him to improve his *programming* skills (he has used the keyword 'programming'). After retrieving the list of LOs that match this query, the ERS executed its recommendations process and got the following results[5]: the content-based recommendation module delivered the LO with ID *LO262*; the collaborative recommendation module proposed a different LO, with ID *LO269*; and finally, the knowledge-based recommendation module delivered again the LO with ID *LO269*.

The ERS selected from these three proposals the LO that should be more relevant for the student learning objectives. The relevance is understood as the suitability of a LO in view of the student's preferences and profile. Therefore, a LO delivered by our ERS can be considered as 'relevant' if it matches the student's learning objectives (determined by the keywords) and profile (his/her learning style, format, language, and learning time preferences). For this example, the traditional hybridization method that our ERS used to date [9] will follow an approach based on set theory and provide the *LO269* to *Jose*, since it has been recommended by two out of the three recommendation modules.

To evaluate recommendation results according to their relevance for the student, we can use the usual precision formula (see Section 6). Therefore, according to our relevance definition, we get the following results:

– LO262 Precision = 1 content-based recommendation
– LO269 Precision = 0 collaborative recommendation and knowledge-based recommendation

[5]For the sake of simplicity, we only provide the top 1 recommendation results of each module.

which shows how the traditional hybridization method failed to deliver the most relevant LO in this case. In fact, although LO269 is *educationally appropriated* (its type fits the user's learning style) and it is *updated* (it has been updated within the last 5 years), it does not meet other user's preferences. It is not *generally appropriated* (its structure is not atomic and its state is not final, which means that it can be a LO under review), not *technically appropriated* (its language and format do not match the preferences of the user), and not *learning time appropriated* (it exceeds the maximum learning time preferred by the user).

Alternatively, our new argumentation-based hybridization method will trigger the rules shown in Table 3 for LO262 and LO269 with their associated probabilities[6].

The collaborative recommendation module was able to find two similar users *'juan'* that liked LO262, and *'pablo'* that liked LO269, but recommended LO269 since *'pablo'* is more similar to the actual user than *'juan'*. These inferences are also encoded in rules $O1_{LO262}$ and $O1_{LO269}$. Similarly, the knowledge-based recommendation module was able to find a LO258 similar to LO269 and another LO274 similar to LO269 that were successfully recommended in the past to the actual user, but LO274 received a highest vote, and hence, LO269 was recommended. These inferences are also encoded in rules $K1_{LO262}$ and $K1_{LO269}$. All these requirements were also met by LO262. However, while for LO262 all literals hold and all rules have an associated probability of 1, some literals do not hold for LO269 (those that represent the unfulfilled user preferences encoded in the content-based rules), which decreases the probability associated with their rules.

Therefore, as the new argumentation-based hybridization method would be able to generate more arguments to justify the recommendation of LO262, the system would succeed in selecting the most relevant LO for this specific user. For instance, with the rule $C1_{LO262}$ the ERS can use the explanation scheme $E1$ (see Table 2 in Section 4) and provide the user with an argument to justify the recommendation of $LO262$: *'The learning object LO262 fits the topic 'Programming', is suitable for your 'auditory' learning style, and it is atomic and stable'*.

[6]Only a selection of these rules are presented for the sake of simplicity.

CONTENT-BASED RULES	$C1_{LO262}$ *P=1: recommend(user, LO262) ← educationally_appropriate(user, LO262) ∧ generally_appropriate (LO262)*
	$C1_{LO269}$ *P=0.5: recommend(user, LO269) ← educationally_appropriate(user, LO269) ∧ generally_appropriate (LO269)*
	$C2_{LO262}$ *P=1: recommend(user, LO262)← educationally_appropriate(user, LO262) ∧ generally_appropriate(LO262)) ∧ technically_appropriate(user, LO262)*
	$C2_{LO269}$ *P=0.33: recommend(user, LO269)← educationally_appropriate(user, LO269) ∧ generally_appropriate(LO269)) ∧ technically_appropriate(user, LO269)*
	$C3_{LO262}$ *P=1: recommend(user, LO262) ← educationally_appropriate(user, LO262) ∧ generally_appropriate (LO262) ∧ updated(LO262)*
	$C3_{LO269}$ *P=0.66: recommend(user, LO269) ← educationally_appropriate(user, LO269) ∧ generally_appropriate(LO269) ∧ updated(LO269)*
	$C4_{LO262}$ *P=1: recommend(user, LO262) ← educationally_appropriate(user, LO262) ∧ generally_appropriate(LO262) ∧ learning_time_appropriate(LO262)*
	$C4_{LO269}$ *P=0.33: recommend(user, LO269) ← educationally_appropriate(user, LO269) ∧ generally_appropriate(LO269) ∧ learning_time_appropriate(LO269)*
COLLABORATIVE RULES	$O1_{LO262}$ *P=1: recommend(user, LO262) ← similarity(user, 'juan') > α ∧ vote('juan', LO262) ≥ 4*
	$O1_{LO269}$ *P=1: recommend(user, LO269) ← similarity(user, 'pablo') > α ∧ vote('pablo', LO269) ≥ 4*
KNOWLEDGE-BASED RULES	$K1_{LO262}$ *P=1: recommend(user, LO262)← similarity(LO262, LO258) > β ∧ vote(user, LO258) ≥ 4*
	$K1_{LO269}$ *P=1: recommend(user, LO269)← similarity(LO269, LO274) > β ∧ vote(user, LO274) ≥ 4*

Table 3

Defeasible rules triggered for LO262 and LO269

6. Evaluation

To test and evaluate our argumentation-based recommendation method, we have added this new hybridization method in the module that performs the hybridization process of our ERS for the FROAC (Federation of Learning Objects Repositories of Colombia) repository. Also, we wanted to test the performance of this method as a new argumentation-based recommendation algorithm. Therefore, we have also implemented a new argumentation-based recommendation module that can operate in parallel with the previous three recommendation modules (content-based, collaborative and knowledge-based). In this section, we present the results of our evaluation tests.

6.1. Evaluation Methodology

To run the evaluation tests, we selected a set of 29 students of the Universidad Nacional de Colombia that were registered as FROAC users. Concretely, we selected pre-graduate students from management systems (55%) and electrical engineering (24%) degrees, graduate students (14%) and PhD students (7%) in systems engineering.

Figure 5 shows the website that the students used in the evaluation test to query the FROAC. If the student is logged on the system, he/she can execute searches by keywords and the recommendation system filters and selects the LOs that match these keywords to apply over them the specific recommendation technique configured in the system.

We used a dataset of 75 LOs from different repositories of FROAC for our evaluation tests. The students were randomly presented with 10 of these LOs and each assigned ratings for them (an average of 8 ratings per student, since some students decided to skip some ratings). These ratings measure how much the student finds appropriate and useful each LO for his/her learning objectives (from 1 -dislikes- to 5 -likes a lot-).

Each LO was characterised by the features specified in the Figure 1 of Section 2. To identify the characteristics of the student profile we took as reference our previous work [29], where the following features were selected:

- Personal information: ID, Name, Surname, Birth Date, Email, Sex, Language, Password, Country, Department, City, Address, Phone.
- Preferences:
    - * Level of Interactivity: students were asked if they prefer LOs that allow interaction or just presentation of content. We got a distribution of 48% of students who prefer LOs with low interactivity level and 52% of students that prefer LOs with high interactivity level.

Fig. 5. Website to query the Federation of Learning Objects Repositories of Colombia

  * Language: 90% of students preferred Spanish LOs (as was expected by their nationality) and only 10% preferred LOs in English.
  * Format: 38% of the students selected *jpeg* as they preferred format, 38% *mp4* and 24% *pdf*.
- Learning Style: we used the *VARK learning styles model* [30] to determine the learning style of each student. Concretely, we got a distribution of 31% Auditory, 7% Kinaesthetic, 10% Reader, and 52% Visual students.
- Usage History: ID of each LO evaluated, Rating, Date of use.

To evaluate the performance of the recommendation system proposed, we removed the ratings of one target student, requested the system to recommend LOs for him/her, and compared these LOs with those that the student actually used and rated. Then, we computed the classical *precision*, *recall* and *F1* scores on the top 5 and top 10 recommendation results [31] and averaged the 29 values obtained, one per student:

$$Precision = \frac{Relevant\ LOs \cap Retrieved\ LOs}{Retrieved\ LOs} \quad (2)$$

$$Recall = \frac{Relevant\ LOs \cap Retrieved\ LOs}{Relevant\ LOs} \quad (3)$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

We performed two types of evaluation tests: *single-module recommendation tests* and *hybridization module recommendation tests*. For the first type, we configured the system to work with a single recommendation module and compared the performance of the argumentation-based recommendation method proposed in this work with the performance of the other recommendation techniques. Therefore, we run the recommendation system without performing any hybridization process (i.e. by only delivering those LOs recommended by using a specific recommendation module working alone). The recommendation modules implemented and evaluated were the following:

- A *random* recommendation module, which provides LOs at random.

- A *content-based* recommendation module, which generates its recommendations by applying inference rules among LOs metadata and the student's learning style.

- A *collaborative* recommendation module, which seeks similar user profiles to deliver items that have been assessed by students with similar profiles.

- A *knowledge-based* recommendation module, which searches some LOs similar to those that the student has previously assessed.

- An *argumentation-based* recommendation module, which implements the method proposed in this work and selects those LOs for which the system is able to generate more support (more arguments and with high probability values).

For the second type of evaluation tests, we configured the system to perform the hybridization process and evaluated the performance of our argumentation-based recommendation method when it works as an hybridization technique. In this case, the content-based, collaborative and knowledge-based recommendation modules were requested to deliver their best 5 (or 10) recommendation results and then, the hybridization module was in charge of combining these results and provide a unique list of 5 (or 10) recommended LOs to the student. Therefore, we compared the performance of several hybridization methods for the hybridization module. Concretely, we implemented: a *random* hybridization method; some of the usual hybridization methods proposed in [6] (i.e. *weighted*, *switching*, *mixed*, and *cascade*); an *intersection-based* hybridization method based on the authors' previous work [29]; and an *argumentation-based* hybridization method proposed in this work. Table 4 shows a recapitulation of the selection techniques that these hybridization methods present. The selection technique shows how the information from each recommender is merged to get a final hybrid recommendations list. For a more complete definition, next we present a deeper explanation of each hybridization method.

- The *Random Hybridization Method* combines at random the top LOs recommended by each recommendation module and thus presents a random list to the user.

- The *Weighted Hybridization Method* provides a score for each list of LOs provided by each recommendation technique. Then, the lists are combined in a unique recommendation list by using a weighting factor that weighs the importance of each recommendation module. We used weighting factors of 0.5 (the median quartile), 0.3 and 0.2 for each one of the three basic recommendation techniques, and computed an average for any possible combination of weighting factors for the techniques (for instance, in the first iteration, we assigned a weighting factor of 0.5 for the content-based technique, 0.3 for the collaborative technique and 0.2 for the knowledge-based technique; in the second iteration, we assigned a weighting factor of 0.3 for the content-based technique, 0.2 for the collaborative technique and 0.5 for the knowledge-based technique; and so on for each possible combination of techniques.). The problem with this method is that the relative value of the weighting factor for the different techniques is uniform and does not consider the amount of information that is available for each technique (amount of known attributes of each object, amount of similar users, amount of past evaluations available, etc.). However, this information can greatly affect the quality of the recommendations provided by each module and hence, should be taken into account to allocate a specific weighting factor for each technique.

- The *Switching Hybridization Method* switches between the recommendation techniques depending on the current situation by using a switching criterion which takes into account the performance of the recommendation modules in the past. Thus, one of the recommendation modules is called first to provide a list of LOs. If it does not deliver appropriate-enough recommendation results, the next module is executed, and so on. The disadvantage of this technique is that if the first recommendation technique provides feasible results, the other techniques are not executed and any user-relevant LOs that can only be retrieved with a specific recommendation technique may be missed out.

- The *Mixed Hybridization Method* presents all the results of all recommendation techniques, ordered randomly. Its main disadvantage is that, since it does not use any criteria to sort the results, the results list can first show LOs with less relevance than others shown in lower positions.

- The *Cascade Hybridization Method* proposes a step-by-step recommendation technique. In the first step, a specific recommendation technique is executed to provide an initial list of candidate LOs to be recommended. In the next step, a second technique refines the recommendation list from the set of candidates, and so on until a final list of LOs is obtained. Thus, each step refines the recommendations of the previous ones. The results of this method are heavily influenced by the specific order in which each recommendation tech-

Hybridization method	*Selection technique*
Random Hybridization Method	Randomises top LOs
Weighted Hybridization Method	Orders top LOs based on scores provided by recommenders and their weights
Switching Hybridization Method	Switches between recommenders based on their performance
Mixed Hybridization Method	Randomises *all* results
Cascade Hybridization Method	Refines the result of one recommender sequentially with the rest of recommenders
Intersection Hybridization Method	Returns the intersection of all recommender results
Argumentation-based Hybridization Method	Returns the more supported items (based on arguments)

Table 4

Selection techniques for hybridization methods

nique is executed.

- The *Intersection Hybridization Method* is based on the intersection operation from set theory and provides such common objects selected by the different recommendation modules. Thus, those objects that have been selected by a greater number of techniques will be shown in the top of the recommendation list. The main advantage of this technique lies in its potential to select the objects that most of the techniques considered most relevant for the student.

- The *Argumentation-based Hybridization Method* executes in parallel the three basic recommendation modules to generate three lists of candidate LOs. Then, it provides an ordered recommendation list with those LOs for which the system is able to generate more support (more arguments and with high probability values). This method has the additional advantage to be able not only to provide a list of recommended LOs, but also to enhance these recommendations with an explanation (an argument) that justifies their suitability for a specific student. Therefore, this technique can also exert a persuasive power that may motivate students to make use of the LOs recommended by the system.

Finally, we performed a proof of concept to evaluate this persuasive power. Therefore, during the tests performed with the argumentation-based hybridization method, the students were presented with the arguments that support each LO recommended by the system and asked if they found these arguments useful and convincing.

6.2. *Single-module recommendation tests*

Figure 6 shows the precision at 5 ($P@5$) values averaged across all test students, as obtained for all recommendation techniques. The boundaries of the boxes denote the 25th and 75th percentiles, and the overall average of the $P@5$ values are marked by the dots inside the boxes. As illustrated in this figure, the argumentation-based recommendation technique outperforms the other techniques on its precision to retrieve relevant objects. In addition, as shown in Table 5 it also gets better results on the number of relevant objects retrieved. This technique makes use of different types of knowledge, represented in different types of defeasible rules, and generates as many arguments as possible to support or reject a recommendation. In this sense, it is able to represent and use the underlying reasoning patterns that the other techniques follow to make inferences, and as expected, this results in better LOs recommended to the student. The knowledge-based recommendation module also achieves good results. This module recommends LOs similar to those that the student used and rated in the past. In our tests, the small size of the LOs dataset will probably make the knowledge-based recommendation technique to recommend exactly the same LOs that the student selected in the past (and potentially found useful). However, the content-based and the collaborative module, which relay on similar items and similar users respectively to provide recommendations, are negatively affected by this same reason.

For the case to provide a list of 10 recommendations, the argumentation-based recommendation technique also gets to better recommendations. The precision value decreases, as shown in Figure 7, mainly because some students rated less than 10 LOs, and those LOs retrieved for which we do not had a rating were counted as false positives. However, as illustrated by the recall value shown in Table 6, the module is able to recommend more relevant LOs.

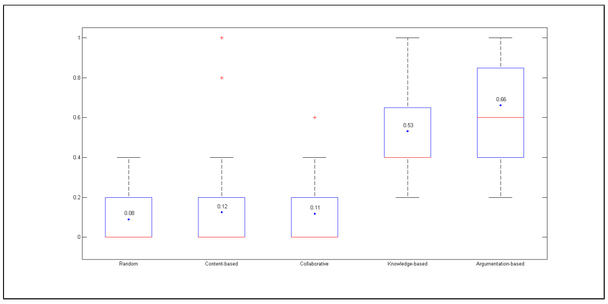

Fig. 6. $P@5$ on the single-module recommendation tests

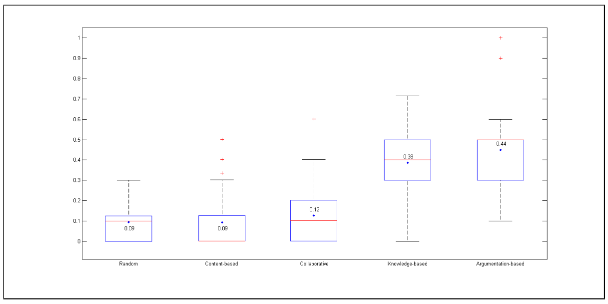

Fig. 7. $P@10$ on the single-module recommendation tests

Top-5 Recommendation Results	*Random*	*Content-based*	*Collaborative*	*Knowledge-based*	*Argumentation-based*
Precision	0.0896	0.1241	0.1172	0.5310	**0.6620**
Recall	0.0630	0.1105	0.1262	0.3836	**0.5138**
F1-score	0.0740	0.1169	0.1215	0.4454	**0.5786**

Table 5

Top-5 Single-module recommendation tests

Top-10 Recommendation Results	*Random*	*Content-based*	*Collaborative*	*Knowledge-based*	*Argumentation-based*
Precision	0.0931	0.090	0.1245	0.3853	**0.4494**
Recall	0.1076	0.1962	0.2173	0.4956	**0.6111**
F1-score	0.0998	0.1241	0.1583	0.4336	**0.5179**

Table 6

Top-10 Single-module recommendation tests

6.3. Hybridization module tests

In the tests that evaluate our argumentation-based method when it acts as a hybridization technique we also achieved good results. As illustrated in Figure 8 and Table 7, the argumentation-based hybridization technique outperforms the others. Overall, all precision and recall values are slightly better than in the case where the recommendation modules act isolated. This is reasonable, since the hybridization process obtains a recommendation list with the best recommendations achieved by all recommendation modules, and hence the drawbacks of a technique can be complemented by the advantages of another. However, the cascade method is heavily influenced by the specific order in which each recommendation technique are executed, and this leads to very bad recommendation results in our settings.

For the case to provide a list of 10 recommendations, as illustrated in Figure 9 the argumentation-based recommendation technique also gets the same good results that in the case of acting as a single recommendation module. This can be explained by the fact that the argumentation-based technique is able to recommend the best possible 10 LOs with our tests configuration and the dataset that we used. Hence, its use as hybridization technique may improve the results of the three basic recommenders acting isolated, but cannot outperform itself as a recommendation method. Also, as shown in Table 8, the mixed hybridization technique is able to provide more relevant LOs than the argumentation-based technique. The reason behind may be that this technique retrieves all results of the three basic techniques and presents them in a random order to the student. In this sense, it tries to make use of all the three inference processes that the recommendation modules use to provide recommendations and in our tests, it happened that its random presentation order put relevant LOs on the top of the 10 recommendations list.

Finally, we evaluated the persuasive power that the arguments generated with the argumentation-based

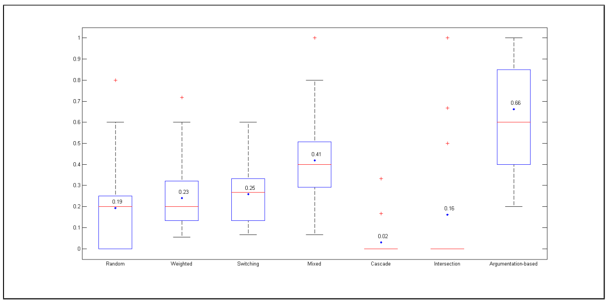

Fig. 8. $P@5$ on the hybrididation module recommendation tests

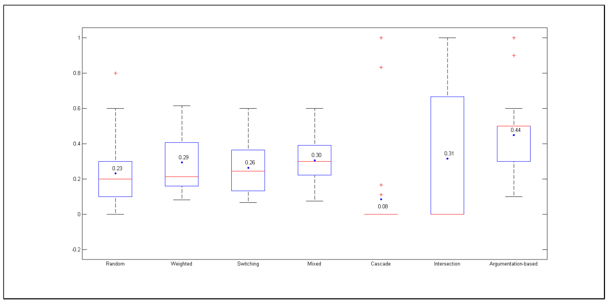

Fig. 9. $P@10$ on the hybrididation module recommendation tests

Top-5 Recommendation Results	*Random*	*Weighted*	*Switching*	*Mixed*	*Cascade*	*Intersection*	*Argumentation-based*
Precision	0.1931	0.2369	0.2556	0.4151	0.0211	0.1609	**0.6694**
Recall	0.1502	0.2541	0.2551	0.3997	0.0219	0.1795	**0.6744**
F1-score	0.1690	0.2451	0.2554	0.4072	0.0215	0.1697	**0.6719**

Table 7

Top-5 Hybridization module tests

Top-10 Recommendation Results	*Random*	*Weighted*	*Switching*	*Mixed*	*Cascade*	*Intersection*	*Argumentation-based*
Precision	0.2310	0.2931	0.2637	0.3057	0.0842	0.3160	**0.4494**
Recall	0.3254	0.2895	0.4413	**0.7058**	0.0379	0.1082	0.6111
F1-score	0.2702	0.2913	0.3301	0.4267	0.0523	0.1612	**0.5179**

Table 8

Top-10 Hybridization module tests

recommendation method can exert in the students. This was a simple proof of concept, where the students were shown arguments associated with the recommended LOs when the system was configured with this recommendation method. Figure 10 shows an example (in Spanish) of the type of textual arguments that our system was able to generate. This argumentation means the following for the user *Angela Maria*:

> *Angela Maria, the following Learning Object {TITLE} is recommended to you because it is educationally appropriate for you since your learning style is "Reader", which corresponds to the learning style of this educational resource. In addition, your kind of interactivity is similar to the interactivity that the learning object provides to its users. The learning object is considered generally suitable for you because its structure is atomic and it is currently completed. The educational resource is updated and the learning time of the resource is appropriate, since it is less than an hour.*

All students provided us with feedback about their feelings with the arguments. Overall, everyone found these arguments useful to understand why the system recommended a set of specific LOs to them, which motivated them to actually use these recommendations. However, the system presented all possible arguments for each recommendation at the same time, and the students found a bit boring to read the entire explanation text. Overall, they considered that the system provided them with too many information. In future work we will analyse how to present these arguments in a more friendly way (e.g. by means of pictures or by selecting only the most relevant argument).

7. Conclusions and future work

This paper has proposed the employment of an argumentation-based formalism for modelling a hybrid recommender system which recommends LOs for specific students. The method was implemented both as an extra recommendation module and as a new hybridization method in an educational recommender system for the Federation of Learning Objects Repositories of Colombia. An evaluation with real students and a reduced dataset from several of these repositories was carried out. In all tests, the proposed argument-based

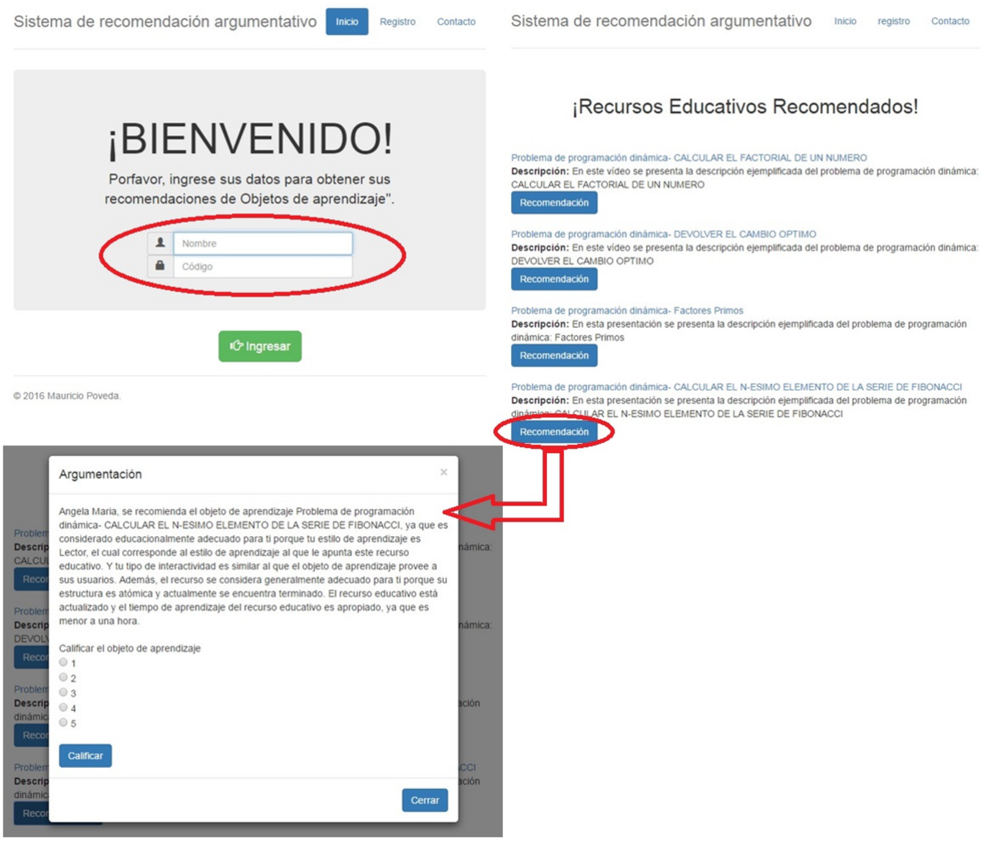

Fig. 10. Example Screenshot of the Execution of the Argumentation-based Hybridization Method in the FROAC Website

method was able to select the most relevant and suitable LOs to recommend. In addition, by using this method, the recommender system can generate arguments to justify its recommendations. Students participating in the tests found these arguments useful to understand the recommendations.

As future work, we plan to enhance the actual simple explanation module with and advanced human-computing interaction module integrated in a conversational agent. The actual persuasive power of the method and its ability to promote changes in the students behaviour must also be tested. In this sense, a comprehensive evaluation must test how students actually enjoy those learning objects that the system has convinced them to use.

Acknowledgements

This work was partially developed with the aid of the doctoral grant offered to Paula A. Rodríguez by 'Programa Nacional de Formación de Investigadores – COLCIENCIAS', Colombia and partially funded by the COLCIENCIAS project 1119-569-34172 from the Universidad Nacional de Colombia. It was also supported by the by the projects TIN2015-65515-C4-1-R and TIN2014-55206-R of the Spanish government and by the grant program for the recruitment of doctors for the Spanish system of science and technology (PAID-10-14) of the Universitat Politècnica de València.

References

[1] Learning Technology Standards Committee. IEEE Standard for Learning Object Metadata. *Institute of Electrical and Electronics Engineers, New York*, 2002.

[2] N.D. Duque, D.A. Ovalle, and J. Moreno. *Objetos de aprendizaje, repositorios y federaciones...conocimiento para todos*. Universidad Nacional de Colombia, 2015.

[3] R. Sikka, A. Dhankhar, and C. Rana. A Survey Paper on E-Learning Recommender System. *International Journal of Computer Applications*, 47(9):27–30, June 2012.

[4] H. Van de Sompel, R. Chute, and P. Hochstenbach. The adore federation architecture: digital repositories at scale. *International Journal on Digital Libraries*, 9(2):83–100, 2008.

[5] H. Drachsler, K. Verbert, O. C. Santos, and N. & Manouselis. Panorama of recommender systems to support learning. *Recommender Systems Handbook*, pages 421–451, 2015.

[6] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modelingand User-Adapted Interaction*, 2002.

[7] A. Klašnja-Milićević, M. Ivanović, and A. Nanopoulos. Recommender systems in e-learning environments: a survey of the state-of-the-art and possible extensions. *Artificial Intelligence Review*, 44(4):571–604, 2015.

[8] P.A. Rodríguez, N.D. Duque, and D.A. Ovalle. Multi-agent system for knowledge-based recommendation of learning objects using metadata clustering. In *Highlights of Practical Applications of Agents, Multi-Agent Systems, and Sustainability-The PAAMS Collection*, pages 356–364. Springer, 2015.

[9] P.A. Rodríguez, D.A. Ovalle, and N.D. Duque. A student-centered hybrid recommender system to provide relevant learning objects from repositories. In Panayiotis Zaphiris and Andri Ioannou, editors, *Learning and Collaboration Technologies*, volume 9192 of *Lecture Notes in Computer Science*, pages 291–300. Springer International Publishing, 2015.

[10] S. Ossowski, C. Sierra, and V. Botti. Agreement technologies: A computing perspective. In *Agreement Technologies*, pages 3–16. Springer, 2013.

[11] S. Heras, V. Botti, and V. Julián. Argument-based agreements in agent societies. *Neurocomputing*, 75(1):156–162, 2012.

[12] N. Manouselis, H. Drachsler, R. Vourikari, H. Hummel, and R. Koper. Recommender Systems in Technology Enhanced Learning. *Recommender Systems Handbook*, 54:479–510, 2011.

[13] M. Salehi, M. Pourzaferani, and S.A. Razavi. Hybrid attribute-based recommender system for learning material using genetic algorithm and a multidimensional information model. *Egyptian Informatics Journal*, January 2013.

[14] A. Zapata, V.H. Menendez, M.E. Prieto, and C. Romero. A hybrid recommender method for learning objects. *IJCA proceedings on design and evaluation of digital content for education (DEDCE)*, 1:1–7, 2011.

[15] V. Vekariya and G.R. Kulkarni. Hybrid recommender systems: Survey and experiments. In *2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP)*, pages 469–473. Ieee, May 2012.

[16] J. Palanca, S. Heras, J. Jorge, and V. Julian. Towards persuasive social recommendation: Knowledge model. *SIGAPP Appl. Comput. Rev.*, 15(2):41–49, August 2015.

[17] G. Linden, J. Hong, M. Stonebraker, and M. Guzdial. Recommendation algorithms, online privacy, and more. *Communications of the ACM*, 52(5):10–11, May 2009.

[18] R. Sinha and K. Swearingen. The role of transparency in recommender systems. In *CHI'02 extended abstracts on Human factors in computing systems*, pages 830–831. ACM, 2002.

[19] P. Dwivedi and K.K. Bharadwaj. e-learning recommender system for a group of learners based on the unified learner profile approach. *Expert Systems*, 32(2):264–276, 2015.

[20] C.E. Briguez, M. Budán, C. Deagustini, A.G. Maguitman, M. Capobianco, and G.R. Simari. Towards an argument-based music recommender system. *COMMA*, 245:83–90, 2012.

[21] C.E Briguez, M. Capobianco, and A.G. Maguitman. A theoretical framework for trust-based news recommender systems and its implementation using defeasible argumentation. *International Journal on Artificial Intelligence Tools*, 22(04):1350021, 2013.

[22] J.A. Recio-García, L. Quijano, and B. Díaz-Agudo. Including social factors in an argumentative model for group decision support systems. *Decision Support Systems*, 56:48–55, 2013.

[23] S. Heras, M. Rebollo, and V. Julián. A dialogue game protocol for recommendation in social networks. In *Hybrid Artificial Intelligence Systems*, pages 515–522. Springer, 2008.

[24] C.I. Chesñevar, A.G. Maguitman, and M.P. González. Empowering recommendation technologies through argumentation. In *Argumentation in artificial intelligence*, pages 403–422. Springer, 2009.

[25] C.E. Briguez, M. Budán, C. Deagustini, A.G. Maguitman, M. Capobianco, and G.R. Simari. Argument-based mixed recommenders and their application to movie suggestion. *Expert Systems with Applications*, 41(14):6467–6482, 2014.

[26] P.A. Kirschner, S.J. Buckingham-Shum, and C.S. Carr. *Visualizing argumentation: Software tools for collaborative and educational sense-making*. Springer Science & Business Media, 2012.

[27] A.J. García and G.R. Simari. Defeasible logic programming: An argumentative approach. *Theory and practice of logic programming*, 4(1+ 2):95–138, 2004.

[28] H. Li, N. Oren, and T.J. Norman. Probabilistic argumentation frameworks. In *Theorie and Applications of Formal Argumentation*, pages 1–16. Springer, 2012.

[29] P.A. Rodríguez-Marín, N.D. Duque-Méndez, and D.A. Ovalle. *Modelo de recomendación adaptativa de objetos de aprendizaje en el marco de una federación de repositorios, apoyado en agentes inteligentes y perfiles de usuario - Tesis de Maestría*. PhD thesis, Universidad Nacional de Colombia Sede Medellín, 2013.

[30] W.L. Leite, M. Svinicki, and Y. Shi. Attempted validation of the scores of the vark: Learning styles inventory with multitrait-multimethod confirmatory factor analysis models. *Educational and psychological measurement*, 70(2):323–339, 2010.

[31] A. Gunawardana and G. Shani. Evaluating recommender systems. In *Recommender Systems Handbook*, pages 265–308. Springer, 2015.