Recommendation of Learning Material through Students´ Collaboration and User Modeling in an Adaptive E-Learning Environment

Daniel Lichtnow[1,3], Isabela Gasparini[2,3,4], Amel Bouzeghoub[4], José Palazzo M. de Oliveira[3], and Marcelo S. Pimenta[3]

[1] Centro Politécnico
Universidade Católica de Pelotas - UCPEL, Pelotas, Brazil
[2] Departamento de Ciência da Computação
Universidade do Estado de Santa Catarina - UDESC, Joinville, Brazil
[3] Instituto de Informática
Universidade Federal do Rio Grande do Sul – UFRGS, Porto Alegre, Brazil
[4] Département Informatique
TELECOM & Management SudParis, Paris, France
{dlichtnow,igasparini,mpimenta,palazzo}@inf.ufrgs.br,
Amel.Bouzeghoub@it-sudparis.eu

Abstract. In this chapter, we present an approach for recommendation of learning materials to students in an e-learning environment. Our aim is to increase the current system's personalization capabilities for students in different scenarios making use of recommendation techniques. The recommendation is produced considering learning materials' properties, student's profile and the context of use. In addition, the process of recommendation is improved through students´ collaboration. In the context of this work, a learning material is a link to a Web page or a paper available on the Web and previously stored in a private repository. The process of collaboration occurs during student's evaluations of the recommendations. These student´s evaluations are used by the system to produce new recommendations for other students. The main features of the recommendations aspects are described and some examples are also used to discuss and illustrate how to provide this personalization.

1 Introduction

A Web-based e-learning environment (ELE) is used by a wide variety of students with different skills, background, preferences, and learning styles. Thus, one of the most desired characteristics of ELEs is being adaptive and personalized [7]. Adaptive educational systems adjust the content presentation and navigation to a student's model. Personalization (or adaptation) is the process of adapting a computer application to the needs of specific users and takes advantage of the acquired knowledge about them. In fact, the use of personalization techniques improves

T. Daradoumis et al. (Eds.): Technology-Enhanced Systems and Tools, SCI 350, pp. 257–278.
springerlink.com

ELE usability since a personalized system customizes the user interface considering the user profile (usually called student model) and each user has a perception that the system was designed specifically for him/her. One aim of our research is to investigate approaches putting the users' profile and contextual knowledge into practice in the development process of real ELEs – in particular of the adaptive environment application for Web-based learning AdaptWeb® (Adaptive Web-based learning Environment) [25], [36] whose goal is to adapt the content, the presentation and the navigation according to the student model. AdaptWeb® is an open source environment and in actual use in different universities.

In the present work we are particularly interested in a strategy for adaptive recommendation of a specific kind of learning objects –Web links to additional information (Web pages, papers, etc). Thus, we use the term *learning material* to make reference to this type of learning objects.

Typically in most of ELEs, a professor (author) suggests learning materials, usually in an unsystematic manual procedure, although some automatic mechanisms have been created. The problem of adequacy evaluation of a suggested Web link (herein called `quality evaluation problem´) appears because most of these learning materials are not adapted to student´s profile, and in many cases could be inadequate to the student in some situations. This may happen for example when the student has not enough knowledge to understand the concepts developed in a learning material suggested as reference, or when the content is written in a language in which the student doesn't have sufficient proficiency.

In these situations, a professor must review the material in advance for each student. In huge e-learning groups, this task is harder and very time-consuming. Thus, it is necessary to create some mechanisms allowing evaluation of learning material and addition information of quality evaluation of learning material.

The most common contents of students' models for e-learning are: students' interests, knowledge, background and skills, experiences, goals, behavior, interaction, preferences, individual traits and learning styles. However, ELEs may be dynamically adjusted not only according to the student's model but also depending on a richer notion of context.

A contextualized ELE provides the learner with exactly the material he needs, and appropriate to his/her knowledge level and which makes sense in a special learning situation, called a scenario in our work [9].

The goal of this chapter is to define an approach to make recommendation of learning materials (Web links or papers in our work) that are most suitable for students´ profile and current tasks (tasks currently being done) in the context of a specific ELE. The present work results of authors' previous experiences with AdaptWeb® [9] [25] [36] and with Recommender Systems [18] [19].

In order to provide a basic summary of recommendation techniques, section 2 presents an overview of aspects related to Recommender Systems. Some recommendations techniques that could be used in the context of an ELE are then identified. In section 3 the architecture and some limitations of actual version of AdaptWeb® are presented. The section 4 describes details about our proposal, that involves the use of an ontology and a collaborative evaluation process to recommend learning materials to students (users of AdaptWeb®). Finally, the section 5 discusses some final remarks and perspectives of future work.

2 Related Works

There is an explosive growth of the volume of information: users should be able to make choices without knowing all alternatives. In this case, user´ expectation is a personalized assistance service - a recommendation. A recommendation looks a sentence like this: "*Customers who bought this CD also bought: Rush – Moving Pictures*". Recommender Systems have been introduced for sifting through very large sets of items selecting those items that are relevant for a determined user.

Recommendations change the way people interact with the Web. Nowadays recommenders help people to choose between diverse products and complex information by providing a more personalized access experience. Recommenders Systems are usually adopted in a great variety of Web sites from e-commerce Web sites like *Amazon.com* to news and information sites like *Digg* and *Slashdot*.

One problem is the complex definition of these recommendation mechanisms due to the lack of semantic representation of the web content. More than this, the quality must be evaluated considering multi-criteria and contextual aspects of the students and can vary from one situation to another for the same student. For example, let´s suppose a student John is studying a new course about Artificial Intelligence (AI). As an activity of AI course, he has to do a survey paper about "search methods". Considering this situation, the system has to select the best learning material in that moment: the material adequate for his particular context (task: do a survey). If we suppose that another student David is trying to understand some basic concepts of AI, the system has to provide totally different materials for him.

A Recommender System is basically a system that try do discover user's interests. The *Tapestry*, considered to be the first Recommender System [28], was created to reduce problems related to the information overload generated by increasing number of electronic messages in a corporate environment. Basically, *Tapestry* filtered the electronic messages based on messages content and in previous evaluation that the first readers have done. Since the system considered the ratings of several users, the process was called *Collaborative Filtering* [13].

Although there are many different approaches to produce the recommendation, *Content-Based* and *Collaborative Recommender Systems* are considered the basic ones [2]. In the *Content-based* approach the user's profile is compared with items to be recommended. In this approach, in general, the user's profile is represented by a set of keywords and the items to be recommended consist of a set of textual documents [4]. The recommendation is produced comparing the user's profiles with documents (the approach employed by many techniques of *Information Retrieval*).

In the *Collaborative* approach the user´s profile is represented by a set of ratings related to specific items that user has previously evaluated. The recommendation is produced comparing the user's ratings to find out which users have a similar profile (Pearson correlation is generally used here). The aim is recommending items to users that others users with similar preferences have evaluated well [2].

It is possible to combine these two approaches to reduce some of problems of each approach [4]. For example, in the case of *Content-based*, the quality of an item is not considered, since it is only based on similarity measures [31]. In the case of *Collaborative* approach, new items will be recommended just after the first evaluation. New users will not be receiving good recommendations since there is not enough information about their profile and preferences (The *cold start* problem) [31].

Many Recommender Systems use users' profiles representation and items semantically poor (e.g. a set of keywords, a set of ratings). In order to make these representations richer and solve some problems present in many Recommender Systems, some researchers have proposed to use ontologies to represent user profiles and items [24] [38] [30] [17]. For example, Vincent Schickel-Zuber and Boi Faltings [30] define a similarity measure used with ontology to reduce problems of absence of ratings related to new items or new users. The idea is that items with similar properties tend to have similar evaluation compared to a previously evaluated item. Thus, a rating given to an item is propagated to new items that belong to the same concept or similar concepts.

Another point is related to the fact that knowledge about users' tasks can allow to produce better recommendations [14] [22] [23]. These users' tasks represent the context of use of an item. Considering the goal of our work there are some interesting proposals regarding to use of Recommender Systems in e-Learning environments. There are different scenarios of traditional Recommender Systems for example, in *e-Commerce*. In their work Drachsler, Hummel, and Koper [8] and Santos [29] discuss aspects related to use of Recommender Systems in *e-Learning* environments. Firstly in an e-Learning environment the recommendation must consider pedagogical aspects rather than just users' preferences (a movie recommender will try to recommend a movie according user's preference). Besides, Drachsler emphasizes that "*the cognitive state of learner and the learning content may change over time and context*". Another point is that beginner's learners could be helped by information given by advanced learners (*What is the best learning material for this task? Which materials my classmates have used before?*).

Considering these aspects, with respect to Recommender Systems and according to [29] "*there is not a single recommendation strategy to apply (due to the diversity of needs and situations)*", we developed an hybrid approach.

In our approach, the process of recommendation combines four aspects: (i) the users' model (e.g. cognitive style, knowledge about language, etc - these data are previously added by system's administrator); (ii) the content and properties of the learning material (Web pages), (iii) the context (tasks related to students), (iv) and finally the quality of recommendation is improved using users' ratings.

Our approach is predestined to undergraduate students. It is not our intent to retrieve automatically material from Web. We believe this is a nice feature suitable for e-learning environments where students have more knowledge and could evaluate the accuracy of the material (some postgraduate courses, for example). In the case where material is retrieved from Web [32], it is important to note that they only use a repository that contains papers that have been previously reviewed by a professor. We believe this kind of materials is not adequate for novice students who are starting to study a subject (especially in the first classes). Thus, in the context of our work, all materials must be reviewed by authors/professors before being available to students.

A mentioned problem of Recommender Systems is related to *cold start* – that means it is not possible to produce recommendation because there is no information about a new user. To solve this problem, some Recommender Systems ask new users to access and evaluate some items. Schickel-Zuber et. al. [30] try to solve or reduce this problem using an ontology. In a similar way, our recommendation process considers user's model, task taxonomy and an ontology to organize learning materials. Our learning materials ontology defines the aspects of quality (quality = "*fitness to use*" [35]) of each learning material. Using this information it is possible to produce recommendation for new users.

Finally, we consider that students can help each other in a collaborative way of recommending material. Students give feedback about their perception of learning materials´ quality. Each rating is related to a specific context- task and it is considered in the student´ profile.

3 AdaptWeb® Environment

The aim of our approach is to improve the personalization process in the context of AdaptWeb® using an ontology and recommendation techniques. This section presents the architecture and the use of the AdaptWeb®.

The AdaptWeb® environment is an adaptive hypermedia system providing the same content adapted to different students groups. AdaptWeb® it is an open source environment in operation on different universities. The Fig. 1 shows the Architecture of AdaptWeb®.

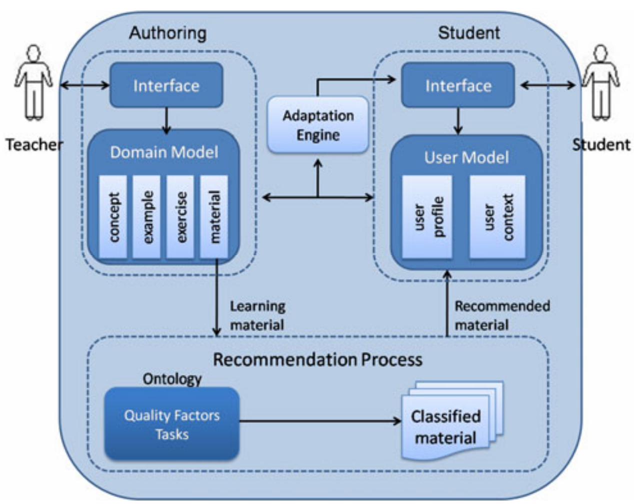

Fig. 1 Architecture of AdaptWeb® with Recommendation Module

Following the architecture, the system is composed of an authoring environment where the professor organizes and creates the structure of content of their courses adapted to degree programs (for example: Engineering, Computer Science or Mathematics), and by an environment for students that produces the personalization.

Thus, using the authoring environment, the professor starts defining the concepts related to each subject. These concepts are organized hierarchically (Fig. 2). This structure is stored in XML (Extensible Markup Language) format in a private repository present in the AdaptWeb®.

Database Systems
1 Introduction to Database Systems
  1.1 The Evolution of Database Systems
  1.2 Overview of a DBMS
  …
2 The Relational Model
  …

Fig. 2 Concepts of Database Course

The authoring environment also allows authors to include learning objects. In this sense, we adopt a learning object broad definition - according to IEEE - as "*any digital entity which can be used, reused or referenced during technology supported learning*"[1].

For each concept there may be a list of examples, exercises and learning materials (Fig. 1). We use the term learning materials to refer to a specific kind of learning objects – the content present on the Web accessible via a Web link (e.g. Web page, papers etc). Fig 3 shows the interface used by professors to include learning materials.

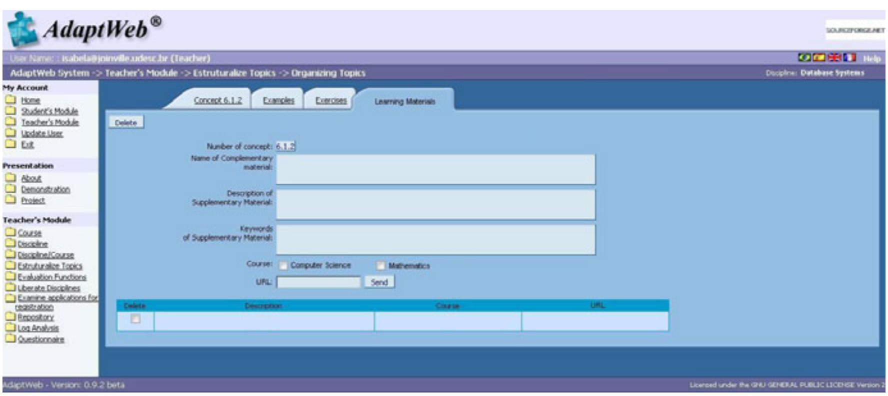

Fig. 3 Interface to include learning materials

[1] http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

Examples and exercises are defined by professors using the authoring environment and are stored in a private repository as XML files. Learning materials are not produced by professors – They only propose Web links that are stored in AdaptWeb® repository.

Regarding to implementation aspects, two DTD (Document Type Definition) were defined for the XML files creation. A DTD depicts the hierarchical structure of concepts and the other DTD describes the specific content of each concept. An algorithm to store the content structured was defined based on DTD. This algorithm creates an XML file for each subject with its respective concepts structure and the features of each concept and an XML file for each concept, which includes tags for concepts, examples, exercises and learning material in its body.

Storing files in XML format makes it possible to structure data in a hierarchical way, because there is always a single XML file with the structure of concepts of the subject and as many XML files as concepts defined. The XML files generation is validated through a parser that scans the documents.

The Adaptation Engine combines the student information with the structure of concepts defined by the professor using authoring environment. In the user model is defined the user profile (e.g. navigation mode preference, the learning style and the language skill) and user context (e.g. the history of navigation, task that user is doing). Student's information are also stored in XML files in the AdaptWeb's repository. The Adaptation Engine generates an instance of an XML file adapted to student and the presentation is dynamically generated. In the interface, the adaptation occurs in the links that are available for the student. For example, if the concept involved in the prerequisites was studied by the student, then the concept that depends on it can be enabled. Thus a concept can assume three categories: 1) "studied", it means that the student has already accessed the concept; 2) "under study", it refers to the concept that is being accessed, also named current; 3) "not studied", a concept can be in this category for two reasons, (a) the concept was not enabled because its prerequisites were not studied yet; or (b) the concept was not studied yet, but it is enabled after prerequisites were studied [36].

Regarding to the navigation mode, the adaptation can work in two ways: tutorial mode or free mode. In the tutorial mode (guide tour), prerequisites criteria among concepts determine the student's navigation, and navigation adaptation is based on the register of concepts studied ("studied", it means that the student has already accessed the concept). In the free mode, the student can study any concept available in the navigation menu. These aspects are presented with more details in some previous publications [25], [36] and [9].

One of the limitations that we observed using AdaptWeb® is the lack of adaptation when proposing learning materials to students. These limitations are due to the fact that learning materials are shown to any student anytime (the only aspect that is considered is the suitable concept). Aspects related to students' profiles and tasks are not considered.

Thus, our proposal focused on the definition of recommendation mechanisms to propose learning materials to students in a more adequately way. As consequence, professors must fill values of learning material's properties (see section 4). In addition, professors must assign each learning material to a specific task, previously identified. A quality measure is also used to classify the learning materials according to quality factors. These aspects, related to recommendation process and techniques, are described in the next section.

4 Integrating Recommendation Mechanism to AdaptWeb®

This section describes our approach to recommend learning materials to students into AdaptWeb®. The recommendation process uses an ontology to classify the learning materials according students' needs and feedback to indicate learning material with more quality to them ("quality" means more appropriate for students). In the present section, initially the ontology that contains learning material's properties and classes is described. This ontology is used to classify a learning material. After, we describe the process of collaboration that takes into account students' evaluations to improve the quality of recommendation. Finally, we discuss about some scenarios of use and aspects related to implementation.

4.1 Multi-Criteria Ontology for Learning Materials

The Recommender System aims to recommend items that are likely to be adequately to users. Quality, in many works, is related to "*fitness for use*" [35]. Our approach defines an ontology that represents quality properties of learning materials. This quality dimensions are described in works that defines some quality content factors. Although there is not an agreement on the quality factors definition, many works agree with Wang and Strong [35] in terms of categories of quality dimensions:

- *Intrinsic*: independent of the user's context, emphasizes that data have quality in their own right.
- *Contextual*: emphasizes that quality must be considered within the context of the task at hand.
- *Representational*: emphasizes aspects related to data format.
- *Accessibility*: emphasizes aspects related to availability and security.

For each of these categories, some quality dimensions must be considered. These quality dimensions are selected considering the requirements of our work and chosen based on their importance considering Bizer [5] and Knight and Burn [15]. Quality dimensions examples related to intrinsic category are Wang and Strong [35] and Pipino, Lee, and Wang [27]:

- *Accuracy*. The extent to which data are correct, reliable, and certified free of error (reflects real world).
- *Objectivity*. The extent to which data are unbiased (unprejudiced) and impartial.
- *Believability*. The extent to which data are accepted or regarded as true, real and credible.

In the context of our work the quality factors related to *Intrinsic* category are assured by authors (the professor). Our assumption is that the author has enough knowledge to select web learning content that have these qualities. Therefore the problem is how to associate a learning material with a context (student and task). This problem is related to *Contextual, Representational* and *Accessibility* dimensions. Contextual quality dimensions considered are [35]:

- *Timeliness*. The extent to freshness of data is appropriate for the task at hand.
- *Relevancy*. The extent to which data are applicable and helpful for the task at hand.
- *Amount of Data*. The extent to which the quantity or volume of available data is appropriate.

Regarding to *Representational* and *Accessibility* categories two quality dimensions are considered:

- *Understandability*. The extent to which data is easily comprehended [27].
- *Accessibility*. The extent to which data is available or easily and quickly retrievable [35].

The ontology represents some classes where the learning materials is classified according to some characteristic. The Fig. 4 shows the classes hierarchy of our ontology. Table 1 presents the properties used in our ontology and the quality dimensions related to each property.

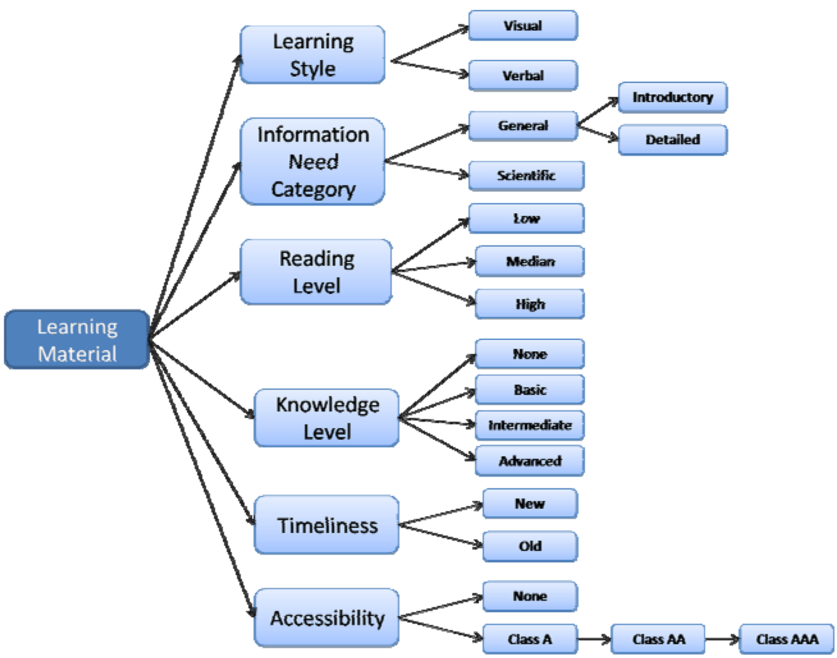

Fig. 4 Ontology´s classes hierarchy

Each Learning Material's subclasses shown in Fig. 4 (*Learning Style, Information Need Category, Reading Level, Knowledge Level, Timeliness and Accessibility*) are related to specific quality aspects. In the ontology *Learning Material*'s subclasses are not disjoints. Consequently, a specific learning material is assigned to each one of subclass. However, the subclasses of these classes (*Learning Style, Information Need Category, Reading Level, Knowledge Level, Timeliness and Accessibility*) are disjoint. Each learning material will be classified according to properties values (see Table 1).

To classify a learning material, in these classes, authors must fill metadata entries before storing a learning material in the repository. Details about how the

Table 1 Learning Material Properties

Property	Description and Quality Dimensions
knowledgeLevelRequired	Refers to a knowledge level (knowledge that a user must to have to understand the content). Can be none, basic, intermediate or advanced. It is related to subject property. Quality Dimension: Understandability.
created	Date of creation of learning material on the Web. It is not related to time when the learning material has been added in AdaptWeb. Quality Dimension: Timeliness
modified	Date on which the learning material was changed on the Web. It is not related to time when the learning material has been updated in AdaptWeb. Quality Dimension: Timeliness
numberOfWords	Can be few or many. Quality Dimension: Amount of Data.
publishedInJournal	Yes or no. Quality Dimension: Understandability.
publishedInConference	Yes or no. Quality Dimension: Understandability.
contentPage	Refers to the type of the content (writing, pictures, graphs, etc.). Quality Dimension: Understandability.
language	Refers to language of the learning material (e.g., English, Spanish, Portuguese). Quality Dimension: Understandability.
readability	Refers to reading ease. It is depending on the language. There are some readability formulas (e.g. SMOG). These formulas estimates the years of education needed to completely understand a text. In the context of AdaptWeb® these property is related to learning material written in a language that is not student's mother tongue. Quality Dimension: Understandability
subject	Refers to some concept of hierarchical structure cited (see Fig. 2). Quality Dimension: Relevancy
textEquivalent	Refers to the fact that the Web page has a text equivalent for every non-text element (e.g., via "alt"). Quality Dimension: Accessibility.
sufficientContrast	Refers to the fact that in the Web pages foreground and background color combinations provide sufficient contrast. Quality Dimension: Accessibility.
acronymExpansion	Refers to the fact that there is a expansion of each acronym in the Web page. Quality Dimension: Accessibility.

professor informs this metadata are presented in section 4.3.1. We present in the following sub-section the ontology's classes, properties and conditions that associate a learning material to specific classes.

4.1.1 Ontology Classes

Knowledge Level. A learning material will be classified in one of subclasses of this class according to students' knowledge level. Thus, the professor must indicate the knowledge level required to use a learning material. For the professor accomplish this, he assigns one specific value to *knowledgeLevelRequired* property. This value can be *none*, *basic*, *intermediate* or *advanced*. According to this value a learning material is associated to a specific class. There are 4 classes (the names of some classes are the same names of values to property):

- *None.* Learning material for students who just start a discipline or a discipline's subject (∀*knowledgeLevelRequired None*[2]).
- *Basic.* Learning material for students with minor knowledge, just attend initial classes (∀*knowledgeLevelRequired Basic*).
- *Intermediate*. Learning material for students that have some knowledge, attend a good percentage of classes (e.g. more than 50%) related to a subject (∀*knowledgeLevelRequired Intermediate*).
- *Advanced.* Learning material for students at the end of course or students that finished a specific subject (∀*knowledgeLevelRequired Advanced*).

Information Need Category. This class is related to idea that for each category of information there are different quality factors and indicators that have to be considered and it is useful to specific demands (based in [12]). The subclasses are:

- *General.* Learning material has not been published in a conference or journal as research paper *(∀publishedInJournal No*) and *(∀publishedInConference No*). This class has two subclasses *Detailed* and *Introductory*. *Introductory* refers to the information whose amount of information is smaller than *Detailed* information Thus a learning material is classified as *Detailed* (∀*numberOfWords Many*) or *Introductory* (∀*numberOfWords Few*).
- *Scientific.* Refers to publications produced specially in academic environments. This kind of publication was published in a Journal *(∀publishedInJournal Yes*) or in a Conference *(∀publishedInConference Yes*).

Learning Style. The learning style is a characteristic defined by the way people prefer to learn. This feature is widely used in adaptive educational systems. A learning-style model classifies students according to how they fit in a number of scales representing how they receive and process information. Several models and frameworks for learning styles have been proposed. The Felder-Silverman's model classifies students according to the way that each one receives and processes the information considering the styles as skills that can be developed [10]. Felder-Silverman's model categorizes students as sensitive/intuitive, visual/verbal,

[2] The Protégé´s ontology representation is used in this paper to describe the ontology.

active/reflective, and sequential/global, depending on how they learn. The visual-verbal dimension of Felder´s learning style model, differentiates learners who remember best what they have seen, e.g. pictures, diagrams and flow-charts, and learners who get more out of textual representations, regardless of the fact whether they are written or spoken [10]. Considering this fact, a learning material can be classified into two classes:

- *Visual.* Learning material that contains pictures or graphs *($\exists$ contentPage some Picture or Graph)*.
- *Verbal.* Learning material that hasn't pictures or graphs *($\forall$ contentPage Text)*.

Timeliness. A learning material is classified as *New* or *Old* according to freshness. Here timeliness is related to data of creation of a learning material, it is not related to date of inclusion of a learning material in environment. This can be useful for some task where it is important to consider the learning material age (e.g. a student is searching about recent advances related to a research area):

- *New.* Refers to learning material that was created or modified recently *($\forall$ created Actual or $\forall$ modified Actual).*
- *Old.* Refers to learning material that was created or modified some time ago *($\forall$ created Outdate and $\forall$ modified Actual).*

It is not easy to know the meaning of *Actual* and *Outdate*. We consider a learning material is actual if it is created or modified in the last year from the date of use. Again is related to date of a learning material, this it is not related to time when the learning material has been added in our environment.

Reading Level. The subclasses are used to indicate the difficulty's degree related to reading. It is important to identify difficulty's degree of learning material written in a language that is not student's mother tongue. For doing this, the author assigns one specific value to *readability* property. This value can be *low*, *medium* or *high*. According to this value a learning material is associated to a specific class like in the case of subclasses of *Reading Level* class (section 4.3.1 discusses details about use of SMOG or Flesh-Kincaid).

Accessibility. The subclasses are used to verify how much a webpage is accessible for users, in W3C terms [33], where we classify a page by none, when a page or a learning material is not accessible, A, when a page has certain proprieties agreeing recommendation, and so on. It is a hierarchy because a Website that fills the requirements of accessibility of *ClassAAA* also fills requirements of *ClassAA*, and *ClassAA* fills requirements of *ClassA*. This accessibility hierarchy can be studied in W3C 1999 [34]. This class is associated to user, not so much with task. Aim to illustrate, the Table 1 contains three properties related to accessibility (*textEquivalent*, *sufficientContrast* e *acronymExpansion*) but there are others. The *textEquivalent* is related to conformance level *A*, *sufficientContrast* is related to conformance level *AA* and *acronymExpansion* is related to conformance level *AAA*.

4.1.2 Taxonomy of Tasks

One important point is to identify which kind of learning material is more suitable to some specifics tasks. Thus, it is necessary to identify tasks and assign each task to ontology's classes. This tasks' list is not exhaustive. Examples of tasks are:

- *Studying Basic Concepts.* At any time (especially in the beginning) a student may need to study or review (in case of doubts) some basic aspects related to a topic. In this case, the student needs learning material that belongs to *Information Need Category -> General -> Introductory* class.
- *Doing a Final Work.* A final work may be writing a program or solving more complex exercises. In this case, the student needs learning material that belongs to *Information Need Category -> General -> Detailed* class.
- *Fulfill a Survey.* This activity requires reading papers about a specific topic. The student needs learning material that belongs to *Information Need Category -> Scientific* class.
- *Search for Recent Advances.* The student needs recent papers about a topic. Therefore, the student needs learning material that belongs to *Information Need Category -> Scientific* class and *Timeliness->New.*
- *Studying for final exam.* Happens generally at the end of a semester or a scholar year. The student needs to review the key subjects and concepts. *Information Need Category -> General -> Detailed* class.

4.2 The Collaboration Process

Although the learning material has been adapted to student's profile and to specific task, the process of recommendation could be improved considering student´s opinion, especially when there is a great number of learning materials.

The evaluation process take in care the context of use: a student gives his opinion about a learning material considering one specific task that he is doing or he has done.

We define a rating function *R* on the space *Student*×*Learning Material*×*Task* specifying how much students liked learning material *lm* on working in a task *t*. Thus, a student evaluate a paper about "XML databases", that he used to *Fulfill a Survey* (see section 4.1.2) as "3". Our approach is similar to Adomavicius and Tuzhilin [1] related to classical *OLAP - Online Analytical Processing* model in databases. Our predicted score for a specific learning material *lm* considering a task *t* is given by (1):

$$\overline{rlmt} = \frac{1}{n}\sum_{i=1}^{n} rlmt_i \tag{1}$$

Where $rlmt_i$ is a rating (range of 0 to 4) given by a student to a learning material (*lm*) considering a specific task *t*. The same learning material can be evaluated with distinct ratings by the same student - one rating for each task. This score is used to generate ranking of learning materials to students in the recommendation process (see section 4.3.3).

In Recommender Systems context, an important point is the collaborative filtering where users who gave similar ratings for the same items are identified. This similarity is measured using Pearson coefficient. After, considering the most similar users (*neighborhoods*) items that do not have rating by a specific user are presumed [2]. This process could be used in our environment, but it seems to be a technique with high computational cost, e.g., if we have only 4 student´s review, we get 6 combinations, but if we have 5 student´s review, the number increases to 10 combinations and so on. The fact of learning materials, tasks and users profiles are richer represented become possible to use another alternative, simpler and with a lower computational cost. Besides, problems related to *cold start* could be decreased.

The techniques described in this section are based in a previous work [11]. Our proposal is also supported by a recent work that shows improvement of search results by users' feedback [3].

4.3 Scenario of Use

The next sections describe the use of the system and some aspects related to its implementation. The start point is the inclusion of new learning materials. After, to explain adaptations provided, we start by describing some learning situations and then we detail how those situations trigger the corresponding adaptation process in AdaptWeb®. Finally the collaborative evaluation of learning material is illustrated.

4.3.1 Including a New Learning Material

The process of content creation is described in the section 3. However, considering the use of ontology in the recommendation process, a new interface for include a learning material must be used. The new proposal interface allows to professor inform data related to ontology (section 4.1).

Regarding to properties presented in Table 1, it could be possible in some cases to extract part of this information automatically [37]. However, sometimes it is too difficult to extract some of these properties' values. This work does not emphasize these aspects, but we have identified some possibilities.

Considering that a learning material could be a Web page, the first problem is that sometimes a Web page may contain other information (e.g. navigation menus, user comments, advertisings, snippet previews of related documents, legal disclaimers etc.). The first step is to identify the main content of a Web page; some recent works address this problem [16].

After identifying the main content of a Web page, it is easy to obtain the number of words and calculate the level of readability. In the case of readability, it is important to consider that Flesh-Kincaid formulas, for example must be adapted to a specific language. In the case of Portuguese language, for example words contain in average a higher number of syllables than English [21].

If there is metadata present following some pattern (e.g. Dublin Core[3]) it is easier to extract properties' values such as "*created*" and "*modified*". However in most of content there isn't any associated metadata, in this case this extract become more difficult, even impossible. In this situation the author must manually enter this information.

It is more difficult to extract other properties' values, such as the subject. In previous works [18] and [19] a technique to identify the text' subject was presented. Following this technique, each subject known in AdaptWeb® is associated to a list of terms and their respective weights.

The presence of terms in a text indicates (with some probability) a specific subject. Weights are used to state the relative importance (or the probability) of the term for identifying the subject in a text (e.g. the term *neural* is associated to *Artificial Intelligence* subject). The relation between concepts and terms is many-to-many, that is, a term may be present in more than one concept and a concept may be described by many terms.

Thus, the text mining method (a kind of classification task) evaluates the relationship between a text and a subject using a similarity function that calculates the distance between the two vectors. One vector represents texts of main content and the other, representing a specific subject, is composed of a list of terms with a weight associated to each term.

The identification of the presence of pictures (*contentPage* properties) could be done by analyzing the *HTML* code of a Web page (*e.g. *). Regarding to language it is possible to consider metadata (if there is) or methods like proposed in Martins and Silva [20]. The most difficult property's value to extract is related to *knowledgeLevelRequired* that refers to the knowledge level that a user must have to understand the content of a subject.

Considering the context of our work, in general, any learning material aims to help students in a specific subject. When they start their study, they need more simplified learning material than when they are deeply studying for a test, for example. Thus, the default value of this property will be *none* for learning material that does not have been published in a conference or in a journal. When the learning material is related to a Journal or a conference the default value of this property will be *high*. For all the other cases, this information must be indicated by the author (professor) in authoring phase.

Regarding the accessibility´s degree over a webpage, using some automatic tool which supports WC3 patters, as *A-Checker*, *A-Prompt*, *Bobby*, *Hera*, *TAW*, *WAVE*, and others [33]. Obviously, this property is related specially to Web pages. Finally, it is possible to extract some other information about Web page using Google API, for example. One example is the snippet that contains a small description about the content of a Web page.

Only the professor can include a learning material. Another point is that the professor can import and export a set of learning materials to use in other moment with other group of students. There is an important restriction here: the concepts structure (e.g. Fig. 2 – section 3.1) must be equal. The possibility of import/export

[3] http://dublincore.org/

learning materials can be usefully to reduce the *cold start* problem, because students can use evaluations of others students who finished their course before.

4.3.2 Using the Ontology to Produce Recommendation

We show some examples in a Database Systems course context where the professor provided a set of links learning materials with diverse content about database system, for example: History and motivation for database systems, Components, DBMS functions, Database architecture and data independence, etc. For a simplification purpose, we have a few variables over student´s model: student’s course and background, student's knowledge, learning style, subject, task, language, language level and country.

In our example, Mike is a computer science student that lives in England and his mother language is English. He is at the end of Database´ course and his task is to fulfill a final work for his grade (*Doing a Final Work*). He has no knowledge in the specific research theme (*XML databases*), but he has good superficial understanding of the overall of Database. One of the course´ tasks is to investigate and use some XML database. Clearly for this, Mike needs a deep understanding of the XML Databases. He has visual learning style, and he has basic language skills in Portuguese.

In this moment, Mike is using AdaptWeb® to access to his task description in order to start. There are some available links learning materials. Only the recommended links are presented to him, according to his profile. Notwithstanding the recommendation module provides user with the best possibilities at this moment, the environment don´t discard the others learning materials links, and if the user wants to see the whole content, he can do it (in “*more materials*”, Fig. 5).

A more rigorous representation of this situation is given as follows (according to notation defined in Eyharabide et. al. [9]):

Situation 1 =
{(Student.Mike, **isStundentof**, *Grade.ComputerScience),*
(Student.Mike, **isCoursing**, *Course.DatabaseSystems),*
(Course.DatabaseSystems, **hasSubject**, *Subject.XMLDatabase),*
(Student.Mike, **isLearning**, *Subject.XMLDataBases),*
(Student.Mike, **hasUserKnowledge**, *UserKnowledge.bad),*
(Course.DataBasesystems, **hasLearningMaterial**, *Language.english),*
(Course.DataBasesystems, **hasLearningMaterial**, *Language.spanish),*
*(Course.DataBasesystems,***hasLearningMaterial**, *Language.portuguese),*
(Student.Mike, **hasUserTask**, *UserTask.finalWork),*
(Student.Mike, **hasStyle**, *LearningStyle.visual),*
(Student.Mike, **hasMotherTongue**, *Language.english),*
(Student.Mike, **hasLanguageSkill**, *Language.english),*
(Student.Mike, **hasLanguageSkill**, *Language.portuguese),*
(Student.Mike, **hasEnglishLanguageLevel**, *LanguageLevel.high),*
(Student.Mike, **hasPortugueseLanguageLevel**, *LanguageLevel.low),*
(Student.Mike, **isCitizenOf**, *Country.England)}*

Thus, the recommender mechanism suggests the material according of these features, and classifies them. First, the recommender removes links of learning material with Database introductory pages. Second, it matches the study subject with the pages contents, eliminating pages with no corresponding subject. It verify the content and the student' learning style.

As Mike has visual learning style (based on Felder and Silverman dimensions), the recommender mechanism is going to prioritize learning material related to this learning style. After, it classifies links more relevant to less relevant. The ordering results are presented in Fig. 5 (The data and the interface are just illustrative). Basically, more relevant items have properties according the context (task and user' profile). Considering Mike's task and profile the more important classes of ontology are *Learning Style -> Visual*; *Information Need Category -> General -> Detailed* and *Knowledge Level->Advanced*. Note that *Timeliness* is not too important here. Regards to *Reading Level*, learning material written in Portuguese will be recommended only with readability is low (*Reading Level->Low*).

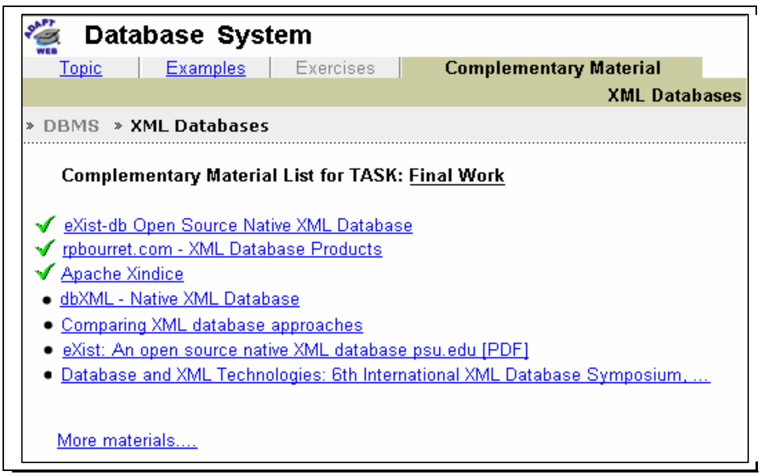

Fig. 5 Learning Material adapted to Mike in situation 1

4.3.3 Collaborative Evaluating of Learning Material

The last aspect of our approach is related to collaborative evaluation of learning materials recommended to students. This process of evaluation starts when a student receives a recommendation that is produced using the ontology (see section 4.3.2).

Thus, regarding to collaborative aspects of our approach, considering the situation 1, Mike will be invited to give a grade for each learning material that he had accessed to perform his task (final work). The data and figures showed here are only to illustrate our approach. The interfaces are being implemented. The process of evaluation will occurs following these steps:

1. Student receives recommendations of learning material (Fig. 6);
2. Student accesses some learning materials (Fig. 6);
3. The AdaptWeb® registers student's access;

4. The AdaptWeb® requires a rating for each accessed learning material. Each rating is represented by a numerical value: 0-Irrelevant, 1-Partial Irrelevant, 2-Neutral, 3-Partial Relevant, 4-Relevant;
5. The AdaptWeb® registers student's ratings.

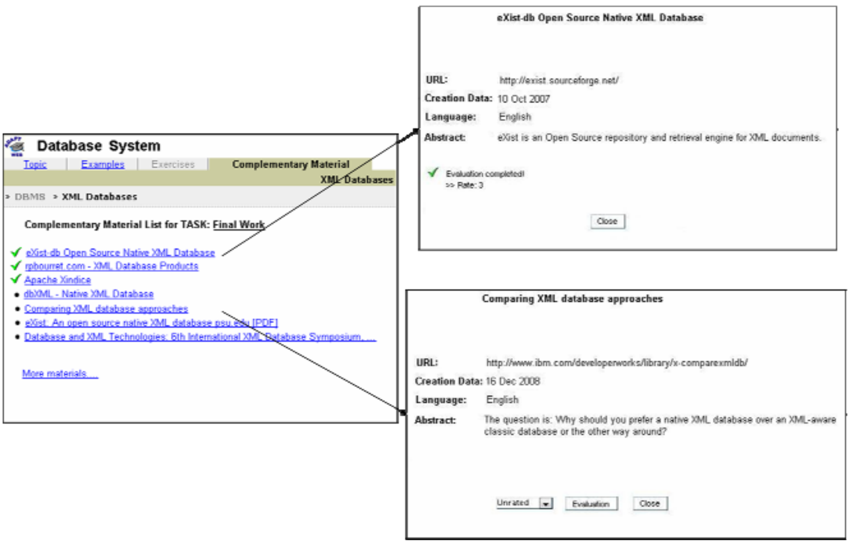

Fig. 6 In situation 1, Mike evaluates the quality of recommendation.

The recommendation process occurs in similar situations to those described in the section 4.3.2 and we combine students' ratings to improve the quality of recommendation. The new recommendation process consists of following steps:

1. The learning materials are recommended as described in the section 4.4.2.
2. The ratings are retrieved. The system considers only the student's ratings related to specific task (see section 4.2).
3. The average rating is calculated (more details in section 4.2).
4. The ranking is generated.

A problem found here is related to *cold start* (see section 2): the learning materials without ratings or with a few ratings will be not recommended to student.

To reduce this problem our approach is to present all learning material to student in a distinguished way into the interface. Thus the system will show learning materials without ratings, with few rating (only 10% of students had evaluated) and the rated ones (the ranking generated using this strategy) in a different manner.

In the case of learning materials without ratings or with few ratings the system will show these learning materials in a random order.

5 Conclusions

As e-learning systems become more sophisticated, new opportunities and new challenges are emerging. One meaningful example is the need to deal with recommendation of diverse learning materials, and in different scenarios.

We proposed the use of ontology to evaluate the quality factor of these learning materials according to different student´s profiles and tasks. This chapter presents how this ontology is integrated in the ELE AdaptWeb®, whose objective is to adapt the content, the navigation and the interface for each student.

In addition, we propose the use of students' evaluation to improve the quality of recommendation. In these sense we are considering some techniques used in the Collaborative Recommender Systems context. Regarding to use of collaborative recommendation techniques, firstly, collaborative filtering techniques, in general, do not consider contextual aspects [2]. Furthermore, the use of items' properties, users' profiles' and task's taxonomy helps recommendation even when there are no ratings - as others works we use an ontology to reduce the *cold start* problem. About *cold start*, we decided to present the learning materials evaluated and learning materials without ratings as well.

Our aim is to improve even more the student´s learning, by giving them the best available learning materials, where the notion of "best" is totally oriented by multi criteria recommendations.

The present work results of experiences of authors with AdaptWeb® and with Recommender Systems. This chapter describes a work that is being developed. We intend to implement and incorporate all these features in the actual version of AdaptWeb®. In addition, our future works include:

- Building more complete task´s taxonomy similar to one present by Broder [6] in the Web Search Goals field. In our case we focus on learning tasks;
- Improving the ontology using others quality metrics, e.g. considering some metrics presents in [26].
- Reducing professors' work. A problem found is related to add new learning material. The professor has to inform all properties values in the authorship phase. Some possible solutions are described in section 4.3.1, but they are not implemented yet.
- Carrying out experiments tests, with a variety of actual students in order to validate our proposal.

Acknowledgments. This work is partially supported by CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brazil and CAPES, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brazil.

References

1. Adomavicius, G., Tuzhilin, A.: Multidimensional recommender systems: A data warehousing approach. In: Fiege, L., Mühl, G., Wilhelm, U.G. (eds.) WELCOM 2001. LNCS, vol. 2232, pp. 180–192. Springer, Heidelberg (2001)

2. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowl. and Data Eng. 17(6), 734–749 (2005), doi: `http://dx.doi.org/10.1109/TKDE.2005.99`
3. Agrahri, A.K., Manickam, D.A.T., Riedl, J.: Can people collaborate to improve the relevance of search results? In: RecSys 2008: Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 283–286. ACM, New York (2008), `http://doi.acm.org/10.1145/1454008.1454052`
4. Balabanović, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Commun. ACM 40(3), 66–72 (1997), `http://doi.acm.org/10.1145/245108.245124`
5. Bizer, C.: Quality-Driven Information Filtering. In: The Context of Web-Based Information Systems. VDM Verlag, Saarbrücken (2007)
6. Broder, A.: A taxonomy of web search. SIGIR Forum 36(2), 3–10 (2002), doi: `http://doi.acm.org/10.1145/792550.792552`
7. Brusilovsky, P., Peylo, C.: Adaptive and intelligent Web-based educational systems. International Journal of Artificial Intelligence in Education 13(2), 159–172 (2003)
8. Drachsler, H., Hummel, H., Koper, R.: Recommendations for learners are different: Applying memory-based recommender system techniques to lifelong learning. In: Proceedings of the EC-TEL Conference, Crete, Greece (2007)
9. Eyharabide, V., Gasparini, I., Schiaffino, S.N., Pimenta, M.S., Amandi, A.: Personalized e-learning environments: Considering students' contexts. Education and Technology for a Better World 302, 48–57 (2009)
10. Felder, R.: Learning and Teaching Styles in Engineering Education. Journal of Engineering Education 78(7), 674–681 (1988)
11. Gheller, L.F.M., Pimenta, M.S.: An aditional user interface layer for search mechanisms (uma camada de interfaces adicional para mecanismos de busca). In: S.B. de Computao SBC (ed.) Proceedings of V Symposium on Human Factors in Computer Systems (IHC 2002), vol. 1, pp. 366–370 (2002) (in Portuguese)
12. Glover, E.J., Lawrence, S., Gordon, M.D., Birmingham, W.P., Giles, C.L.: Web search—your way. Commun. ACM 44(12), 97–102 (2001), doi: `http://doi.acm.org/10.1145/501317.501319`
13. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Commun. ACM 35(12), 61–70 (1992), doi: `http://doi.acm.org/10.1145/138859.138867`
14. Herlocker, J.L., Konstan, J.A.: Content-independent task-focused recommendation. IEEE Internet Computing 5(6), 40–47 (2001), doi: `http://dx.doi.org/10.1109/4236.968830`
15. Knight, S., Burn, J.: Developing a Framework for Assessing Information Quality on theWorld Wide Web. Informing Science: International Journal of an Emerging Transdiscipline 8, 159–172 (2005)
16. Kohlschütter, C., Nejdl, W.: A densitometric approach to web page segmentation. In: CIKM 2008: Proceeding of the 17th ACM Conference on Information and Knowledge Management, pp. 1173–1182. ACM, New York (2008), doi: `http://doi.acm.org/10.1145/1458082.1458237`
17. Lee, T., Chun, J., Shim, J., Lee, S.G.: An ontology-based product recommender system for b2b marketplaces. Int. J. Electron. Commerce 11(2), 125–155 (2006), doi: `http://dx.doi.org/10.2753/JEC1086-4415110206`

18. Loh, L., Lichtnow, D., Kampff, A.C., de Oiveira, J.P.M.: Recommendation of complementary material during chat discussions. Knowledge Management & E-Learning 2(4) (2010) (to be appear)
19. Loh, S., Garin, R.S., Lichtnow, D., Borges, T., Rodrigues, R., Simões, G., Amaral, L., Primo, T.: Analyzing web chat messages for recommending items from a digital library. In: ICEIS, vol. (4), pp. 41–48 (2004)
20. Martins, B., Silva, M.J.: Language identification in web pages. In: SAC 2005: Proceedings of the 2005 ACM Symposium on Applied Computing, pp. 764–768. ACM, New York (2005), doi: http://doi.acm.org/10.1145/1066677.1066852
21. Martins, T., Ghiraldelo, C., Nunes, M., Oliveira Jr, O.: Readability formulas applied to textbooks in brazilian portuguese. Notas do ICMSC-USP, Série Computação 28, 11 (1996)
22. Mcnee, S.M., Konstan, J.A. (Adviser): Meeting user information needs in recommender systems. Ph.D. thesis, University of Minnesota, Minneapolis, MN, USA (2006)
23. McNee, S.M., Kapoor, N., Konstan, J.A.: Don't look stupid: avoiding pitfalls when recommending research papers. In: CSCW 2006: Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work, pp. 171–180. ACM, New York (2006), doi: http://doi.acm.org/10.1145/1180875.1180903
24. Middleton, S.E., Shadbolt, N.R., De Roure, D.C.: Ontological user profiling in recommender systems. ACM Trans. Inf. Syst. 22(1), 54–88 (2004), doi: http://doi.acm.org/10.1145/963770.963773
25. de Oliveira, J.P.M., de Lima, J.V., Gasparini, I., Pimenta, M.S., Brunetto, M.A.C., Proença Jr., M., Faggion, R.: Adaptive multimedia content delivery in adaptweb. In: XIII Taller Internacional de Software Educativo TISE, vol. 4, pp. 23–39 (2008)
26. Olsina, L., Rossi, G.: Measuring web application quality with webqem. IEEE MultiMedia 9(4), 20–29 (2002), doi: http://dx.doi.org/10.1109/MMUL.2002.1041945
27. Pipino, L.L., Lee, Y.W., Wang, R.Y.: Data quality assessment. Commun. ACM 45(4), 211–218 (2002), doi: http://doi.acm.org/10.1145/505248.506010
28. Resnick, P., Varian, H.R.: Recommender systems. Commun. ACM 40(3), 56–58 (1997), doi: http://doi.acm.org/10.1145/245108.245121
29. Santos, O.C.: A recommender system to provide adaptive and inclusive standard-based support along the e-learning life cycle. In: RecSys 2008: Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 319–322. ACM, New York (2008), doi: http://doi.acm.org/10.1145/1454008.1454062
30. Schickel-Zuber, V., Faltings, B.: Inferring user's preferences using ontologies. In: AAAI 2006: Proceedings of the 21st National Conference on Artificial Intelligence, pp. 1413–1418. AAAI Press, Menlo Park (2006)
31. Shardanand, U., Maes, P.: Social information filtering: algorithms for automating "word of mouth". In: CHI 1995: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 210–217. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (1995), http://doi.acm.org/10.1145/223904.223931
32. Tang, T., McCalla, G.: Evaluating a smart recommender for an evolving e-learning system: A simulation-based study. In: Tawfik, A.Y., Goodwin, S.D. (eds.) Canadian AI 2004. LNCS (LNAI), vol. 3060, pp. 439–443. Springer, Heidelberg (2004)

33. W3C: Complete list of web accessibility evaluation tools WAI (Web Accessibility Initiative) (2010), http://www.w3.org/WAI/RC/tools/complete
34. W3C: Web Content Accessibility Guidelines 1.0 (1999), http://www.w3.org/TR/WAI-WEBCONTENT
35. Wang, R.Y., Strong, D.M.: Beyond accuracy: what data quality means to data consumers. J. Manage. Inf. Syst. 12(4), 5–33 (1996)
36. Bry, F., Kraus, M.: Adaptive hypermedia made simple with HTML/XML style sheet selectors. In: De Bra, P.M.E., Nejdl, W. (eds.) AH 2004. LNCS, vol. 3137, p. 472. Springer, Heidelberg (2004)
37. Warpechowski, M., Souto, M., de Oliveira, J.P.M.: Techniques for metadata retrieval of learning objects. In: SW-EL International Workshop on Applications of Semantic Web Technologies for E-Learning, with AH 2006. Citeseer, Dublin (2006)
38. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: WWW 2005: Proceedings of the 14th International Conference on World Wide Web, pp. 22–32. ACM, New York (2005), doi: http://doi.acm.org/10.1145/1060745.1060754