A Personalized Recommendation Framework based on CAM and Document Annotations

Julien Broisin\textsuperscript{a}, Mihaela Brut\textsuperscript{a}, Valentin Butoianu\textsuperscript{a}, Florence Sedes\textsuperscript{a}, Philippe Vidal\textsuperscript{a}

\textsuperscript{a}Institut de Recherche en Informatique de Toulouse, 118 route de Narbonne, 31062 Toulouse Cedex 4, France

Abstract

This paper presents a solution for recommending documents to students according to their current activity that is tracked in terms of semantic annotations associated to the accessed resources. Our approach is based on an existing tracking system that captures the user current activity, which is extended to build a user profile that comprises his/her interests in term of ontological concepts. A recommendation service is elaborated, implementing an algorithm that is alimented by Contextualized Attention Metadata (CAM) comprising the annotation of documents accessed by learners. The user profile is updated as soon as an activity is completed; thus, recommendations provided by the service are up-to-date in real time. The original aspect of this recommendation approach consists in combining a user activity tracking system with the exploitation of the semantic annotations associated with resources.

Keywords: Personalized recommendation; attention metadata; semantic web; annotation-based algorithm; ontology-based modeling

1. Introduction

In the context of e-learning 2.0 age, the personalization of the e-learning process becomes essential. Among the multitude of personalization techniques, the recommendation of relevant resources to the users, according to their particular profile, has gained a great importance due to the huge number of resources available across the Internet. The personalized recommendations of resources help users to reduce time for browsing and searching, as well to recognize and even to discover the resources that are of interest for them.

To achieve this goal, semantic Web technologies must be considered. Indeed, these techniques provide support to enhance the traditional recommendation techniques. In order to benefit from these techniques into an e-learning environment, it is necessary to harmoniously integrate their specific technologies. This paper thus presents a solution for recommending documents to students according to their current activity that is tracked in terms of the semantic annotations associated to the accessed resources. Our approach is based on an existing Contextualized Attention Metadata (CAM) framework that captures users’ current activities, and extends this framework to build a user profile that comprises his/her interests in term of ontological concepts. A service dedicated to recommendation

\* Corresponding author. Tel.: +33 561 557 402; fax: +33 561 556 847.
\textit{E-mail address:} broisin@irit.fr.
is elaborated and implements an algorithm that is alimented by Contextualized Attention Metadata (CAM) stored into a CAM repository and comprising the annotation of documents accessed by learners; after the recommendations, activities operated by users are gathered from the learning environment and transmitted to the CAM environment so that the user profile is updated as soon as an activity is completed. Thus, recommendations provided by the service are up-to-date in real time. The original aspect of this recommendation approach consists in combining a user activity tracking system with the exploitation of the semantic annotations associated with resources.

The existing CAM framework is briefly introduced in the next section; details can be found in [3]. Section 3 discusses the user and the document models adopted in the recommender systems, and further exposes a solution for extending the model of our framework in order to meet the requirements of a recommender system focused on ontology-based resources modeling. Then we present our recommendation framework, which comprises the new models and integrates a recommendation service implementing an annotation-based algorithm; before presenting this recommendation algorithm, the current algorithmic approaches are overviewed. Then some conclusions and research perspectives are presented at the end of the paper.

2. The Basis for our Recommendation System

In order to track heterogeneous systems, resources and users implied within a web-based learning environment, we developed a model driven approach that focuses on learning objects/repositories, curriculums/learning management systems, users and activities they process on these entities. To capture CAM defined within these models from learning environments, we set up a distributed architecture that clearly separates the learning and the CAM environments thanks to a Service Oriented Architecture. Fig. 1 focuses on users and resources, but the details of the whole model can be found in [3]. The modeling of resources (TEL_Resource) enables to specify if a learning resource is conformant to e-learning standards. As could be noticed on Fig. 1, a relation association (TEL_IdentityOnResource) links a resource with a user (CIM_Identity) in order to easily specify activities that can be performed by users on resources; the Activity model representing these actions (consultation, download, rating, etc.) does not appear on Fig. 1. The user model is precisely detailed in [14] and includes the IMS-LIP and IEEE PAPI standards together with some additional information describing users’ characteristics: interests, knowledge (TEL_LearnerCognitive), preferences (TEL_LearnerPreferences) or metacognitive profile (TEL_Metacognitive). The whole modeling is characterized by a high abstraction level, and offers the opportunity of defining specific models according to specific objectives.

The support architecture comprises the learning environment on one hand, the CAM framework on the other hand, and an intermediate layer allowing for communication between these two heterogeneous systems; this architecture is depicted on Fig. 2. Therefore, CAM information produced by learners when they perform an activity within a learning system is forwarded to the middleware layer and then stored within a CAM repository. Thus, CAM information can easily be shared and reused by any other web-based system.

However, in order to meet the requirements of a recommendation system like that mentioned in Introduction, where documents are recommended to users according to their recent activity, there is a need for extending the framework:

− The model has to be extended to take into account semantic annotations that are associated with learning resources in order to integrate recommendation functionalities that consider the user’s topics of interests; it will be presented in the next section.
− An additional service has to be designed so that the CAM tracking framework integrates the recommendation algorithm. The treatment of this service is presented in section 4.

3. Extending the Model
In order to design personalized functionalities such as recommendations for different user types, an adaptive system should first establish a user model to represent users’ profile, as well as a document model to organize and classify the resources accessed by users. Moreover, in order to enrich their semantics, such models should be expressed in terms of ontology concepts. We will present further the user and document models adopted in our framework together with the resulting extensions operated on our existing models mentioned in the previous section.

3.1. The User Model

The adaptive hypermedia systems adopt a feature-based modeling technique of the user profile, considering some important characteristics of the user as an individual: knowledge, interests, goals, background, and individual traits [4]. Three solution types were defined for modeling the user profile, based respectively on a set of keywords, on a specific semantic network, or on a set of concepts belonging to multiple existing semantic networks, which could be taxonomies, topic maps, or even ontology [7]. Thus, in the semantic Web community, some approaches defined XML, RDF or OWL versions of the ACM, ODP (Open Directory Project), or ECDL (European Computer Driving Licence) taxonomies1 in order to express user profiles in terms of learning competences in the computer science domain. Moreover, in the particular case of recommender systems, the user profile focuses on user navigation activity, considered in terms of items, pages, (annotated) documents, etc.

Since we are interested in modeling the user goals, we chose the ACM taxonomy to express the learning objectives a user is reaching during his current activity. Our current user model integrates a Goal property (see Fig. 1), but it aims at specifying personal objectives and aspirations, as defined by the IMS LIP standard. The same fact applies for the Interest property of the TEL_LearnerPreference class: it consists of “descriptions of hobbies and other recreational activities”.

Therefore, we introduced the InterestTopics property to the TEL_LearnerPreference class in order to represent the current users’ concepts of interest (see Fig.1). InterestTopics is an array of real numbers; the size of the array matches with the number of concepts specified in the ACM ontology. Depending on how relevant is a concept for the current user goals, each element of the InterestTopics property varies from 0 to 1. Moreover, to meet the specificities of recommender systems, each InterestTopics’ element evolves in real time according to the users’ activities performed on the resources of the learning system they are interacting with (see Section 4); elements’ values are deduced from annotations associated with the resources and specified in the document model.

3.2. The Document model

In e-learning systems, ontology could be used to exclusively annotate materials or in combination with e-learning standards [1]. Various relation types [9] were adopted in order to refine the ontology-based annotations of the Learning Objects (LOs). In recommender systems, documents are treated mostly as items as a whole, or as pages with certain structures, and they are automatically processed in order to develop the document model.

Some existing techniques for document annotation according to a domain model are inspired by the classic Information Retrieval (IR) or by Web Information Retrieval, exploiting the hypertext features, such as page hyperlinks and HTML general tags. The progress from a term-based to a concept-based document indexation was possible due to the latent semantic indexing technique [16] or to some knowledge representation models and methods that are typical to artificial intelligence domain (such as neural networks, semantic networks, Bayesian networks) [11].

In this context, we developed and integrated into our CAM framework a document model expressed through ACM ontological constructs. In our proposed recommendation technique, we take advantage of a previously reported document indexation technique [5] that enables to obtain the ontology-based annotations for textual documents. Thus, the recommendation technique considers the associated annotations for each document visited by the user in order to locate ontology concepts that express the current goal of the user: the profile of each user is updated according to the annotations associated to the visited documents.

The mentioned technique combines the matrix singular decomposition method with WordNet based keywords

1 http://www.acm.org/about/class/, http://www.dmoz.org/, http://www.ecdl.org/
processing. Given a document collection and an ontology, this technique provides, as result, a matrix \( D_{nxm} \) where:
- each row corresponds to a document \( i \);
- each column corresponds to a concept \( j \);
- \( D[i,j] \) represents the weight of the concept \( j \) for the document \( i \), \((i=1,n; j=1,m)\).

To take into account the semantic annotations of learning resources into our CAM models, the Annotations property has been added to the TEL_Resource class (see Fig.2). The matrix \( D \) representing annotations and described above is defined as an array of real numbers; the size of the array matches with the number of concepts specified in the ACM ontology. Depending on how relevant is the concept for the resource, each element of the Annotations property varies from 0 to 1. Being an abstract class, all inheriting resources (e.g. classes representing SCORM modules, courseware, etc.) will include the Annotations property.

Section 4.3 details the recommendation algorithm that takes advantages of the user and document models presented here. Before, we present the global recommendation architecture and a brief state of the art about recommendation algorithms.

Fig. 1. An extract of the CAM model: resources and users

4. The Recommendation Framework

4.1. The Recommendation Service

Fig. 2 illustrates our global architecture based on the WBEM standard and integrating the new recommendation service. It is divided into three parts:
- The first one represents the users’ environment and contains the tools that the users interact with. These
applications play two distinct roles: (1) they represent a rich source of attention metadata (users’ interactions with the applications) and (2) they serve as a medium to deliver recommended resources. Each application contains an agent which collects users’ traces and sends them to the tracking service so that they are stored within the CAM repository on one hand, and listens on a specified port to receive recommended resources from the recommendation service on the other hand. We distinguish here two types of applications: (1) “online” applications (e.g. Moodle, AriadneFinder) are used as a recommendation medium only when the user is online (e.g. when he’s connected to the application), and (2) “offline” applications (e.g. RSS feeds, e-mail service) used to deliver recommendations when the user is offline (e.g. when he’s not connected to any web application).

- The CAM environment is composed of two components: a CAM repository is responsible for storing attention information, whereas a manager is able to manipulate traces stored into the repository.
- The intermediate layer between the user and the CAM environments offers an easy access to the CAM repository. The services of this layer bridge the gap between the user contexts and the tracking environment. Thus, learning (or non-learning) tools are able to easily provide and/or retrieve CAM stored into the repository. The middleware layer contains three services: (1) the model management service is responsible for managing the model classes (e.g. adding new properties or classes, etc.), (2) the tracking service makes it possible to insert new CAM information into the repository (through the manager) and to retrieve these CAM, and (3) the brand-new recommendation service. Since the first two services have been presented in [3], we focus here on the recommendation service.

![The global recommendation architecture](image)

The recommendation service communicates on the right side with the CAM manager in order to access the CAM repository, and with the agents listening for recommendations on the left side. This service is configured to receive notifications whenever a new resource, user or activity instance is created within the CAM repository. Therefore, if a new resource is available, it is recommended to concerned users. When a user performs an activity within a learning system, the resulting CAM data are forwarded to the tracking service in order to be inserted into the repository. Two possible cases are considered depending on the existence of the accessed resource into the repository:

- The resource doesn’t exist. In that case, the creation of instances of both TEL_Resource and TEL_ResourceActivity subclasses is notified to the recommendation service. The recommendation is triggered not only to the user that carried out the activity, but also to other users that may be interested in the current resource. The recommendation comprises a list of resources that matches with current interests topics (InterestTopics property) for the user that carried out the activity, and only the new resource for the other users. By querying the repository, the recommendation service identifies the users that may be interested in
the new resource, as well as the applications where they are currently logged in, thus providing the new resource to concerned agents that are listening for recommendations. If a user is offline the service will use “offline” medium (e.g. RSS feeds or e-mail services) to deliver recommendation.

- The resource exists. Here, only an activity instance is created and then notified to the recommendation service. The recommendation is triggered only to the user that carried out the activity.

Our service is independent from both target learning systems and tracking service: (1) it is not related to one specific system, it can serve various applications, since the recommended content is delivered to listening agents; (2) the moment when a new document is available is not signaled by the tracking service (in this case, the recommendation service would be integrated within the tracking service, depending on it), but is signaled by the manager, allowing our service to be an independent module. Furthermore, the recommendation is performed as soon as a new resource appears into the system, and not afterwards. Moreover, the recommendation is made only to the current user, but also to the users that may be concerned by the new resource. Finally, it provides two modes of recommendations: online and offline.

The algorithm that implements the mechanisms presented above is presented in the next section, after a brief overview of the main recommendation techniques adopted by adaptive hypermedia systems and enhanced by semantic web technologies.

4.2. Recommendation Algorithms: a State of the Art

Two main techniques were developed in the area of adaptive hypermedia systems in order to recommend the most suitable resources to users [6]:

- Content-based recommendation: exploits information derived from document contents, correlating it with the user profile information.

- Collaborative recommendations (social information filtering): the ratings of the similar users are considered to choose the recommended resources, where the similarity is measured on the basis of the user profile values.

A significant enhancement of the recommender systems is brought by the Semantic Web techniques, as they provide the Web resources (users and documents) with a level of comprehensibility for computer applications, by associating them a machine-processable semantics, such as ontology-based annotations. Thus, the use of a wide variety of reasoning techniques is made possible, which could be applied for enhancing the personalization quality.

Inside an e-learning system, in order to recommend certain learning objects to a specified user, a representation of the user’s goals is necessary (as our modeling approach provides), and a supplementary reasoning process which to correlate these with the knowledge about the learning objects provided by the semantic annotations. For such a process, the goal-directed reasoning techniques seem particularly suitable [2].

Various systems integrated an ontological layer in user profiling in order to provide recommendations, such as the CourseAgent [8] or the Foxtrot [12] systems. The last uses a machine learning technique for classifying the research papers according a paper topic ontology, and the user interest profile is developed in term of ontology concepts according to browsed and visited papers, as well as to his feedback.

Because the considered user and document models concern the resources content (expressed through ontological descriptors), we adopt a content-based recommendation technique, while respecting the following functioning principle [13]:

- User profile is developed based on the his/her interactions history with the system;
- At each step performed by the user, a classification algorithm is used in order to predict which resources (not yet accessed) could be of interest for the user.

Among the most popular classification algorithms adopted in the user model development, it could be mentioned: the decisional trees, the k nearest neighbors (km) algorithm, the Rochio algorithm, the probabilistic approach based on Bayesian networks [13][2]. The km algorithm is convenient for the case where the resources are implicitly or explicitly marked by the textual descriptors (as the case of our modeling approach). Their goal is to determine the “nearest neighbors” of each resource, by using a similarity function. Based on these techniques, we built a recommendation algorithm focusing on document annotations and users’ conceptual navigation.

4.3. Our Recommendation Algorithm
The particularity of our recommendation approach consists in supervising the user conceptual navigation through the ontology instead of his/her site navigation: for each visited document, its annotations are considered to define the user current goals as ontology concepts. While the entire information about the considered ontology is stored in OWL, two matrix are used for representing the users’ and documents’ collections:

- \( D_{nxm} \), where \( D[i,j] \) represents the weight of the concept \( j \) for the document \( i \), \( i=1,n, j=1,m \), as mentioned earlier;
- \( U_{pxm} \), where \( U[k,j] \) is the weight of the concept \( j \) in the profile of the user \( k \).

Our framework enables to capture CAM data resulting from the consultation of a document by a user, whatever the type of the accessed resource is. It means that the following recommendation algorithm could be adopted in a general case to recommend any type of resources:

1. At the beginning of his/her working session, in the profile of each learner \( j \) the weight associated to the concepts are null: \( U[i,j] = 0 \), \( 1 \leq i \leq n \);
2. When the user access a document \( k_0 \), the weights associated to the ontology concepts in the model of this document are incremented to the user profile: \( U[i,j] += D[i,k_0] \), \( 1 \leq i \leq n \);
3. First, a documents filtering is accomplished according to the user profile: the concept \( c[i_0] \) is selected having the maximal weight in the user profile and in the model of the document \( D[k_0] \): \( 1 \leq i \leq n \) et \( D[i_0,k_0]=\max\{D[i,k_0], 1 \leq i \leq n\} \). Further, only the documents \( k' \) having \( D[i_0,k'] \neq 0 \) are considered. Thus, user localization is accomplished with respect to his conceptual navigations inside the ontology.
4. Among the considered documents, the nearest neighbors of the document \( D[k_0] \) visited by the user are determined by using the similarity function specific to the \( k \) nearest neighbors algorithm [15]. The cosine similarity between the document \( D[k_0] \) and a document \( D[j] \) is

\[
\text{sim}(D_{k_0},D_{j}) = \frac{D_{k_0} \cdot D_{j}}{|D_{k_0}| |D_{j}|}
\]

5. As reaction to these recommendations, the user will choose to visit another document, and two possibilities could occur. If the chosen document belongs to the list of recommended documents, the algorithm re-iterates the second step: the user chose something that was proposed to him.
6. If the document chosen by the user does not make part of the list of recommended documents, it could means that the user changed the focus of his/her goal. However, in the development of the user profile, we still maintain a certain degree of importance for the previous focused concept alongside with those granted to the new one. For determining the new concept \( c[i_1] \) that expresses this goal, the technique described at the step 3 is re-applied. Further, the similarity between \( c[i_0] \) and \( c[i_1] \) will be computed according to the formula provided by [10], that is proved to be the best similarity measure between two concepts belonging to a semantic network:

\[
\text{simTopic}(i_1; i_2) = \begin{cases} 
\frac{e^{-\alpha l} \cdot e^{-\beta h} - e^{-\alpha l} \cdot e^{-\beta h}}{e^{-\alpha l} + e^{-\beta h}} \text{ if } i_1 \neq i_2 \\
1, \text{ otherwise}
\end{cases}
\]

where \( l = \) the length or the shortest path between \( c[i_1] \) and \( c[i_2] \),
\( h = \) the number of layers between \( i_1 \) and \( i_2 \),
\( \alpha = 0.2, \beta = 0.6 \) are proved as being the optimal values.
7. This similarity function is sub-unitary and will be applied to the user profile in order to reduce its weight (and to increase the importance of the new document selected by the user). At this point, the second step is reiterated.

While being applicable to any type of learning resources, our approach has the advantage of being independent from the navigational structure of a particular system: recommendations are based on annotations associated with documents, and modeled in a uniform way within both the user’s profile and the resource’s description. The next section illustrates the case where a learner accesses different learning objects materialized as online documents. Some relevant documents are recommended to him/her, according to the interest topics depicted by the annotations.
associated with the accessed documents.

4.4. Recommending Resources to Learners

The recommendation process has to be performed not only when the users are online, but also when they are inactive. Thus, we identified two communication modes to deliver personalized recommendation: synchronous and asynchronous delivery.

Fig. 3. Synchronous and personalized recommendation through the Ariadne Finder

Synchronous delivery, or real time delivery, is performed when the user is online, that is when the user is connected to at least one learning application; to receive recommendations, the target application has to integrate an agent that listens to requests sent by the recommendation service.

We integrated such an agent into the Ariadne Finder (see Fig. 3), a web application that allows searching for learning objects stored into the GLOBE repositories. The user interface of the Finder is divided into three main parts: the frame on the left contains the documents accessed by the logged-in user, the main frame relates the GLOBE documents matching with his/her search criteria, and the frame on the right presents a list of recommended documents stored into the CAM repository. Let us note that the content of this last frame evolves not only when the connected user performs an action on a document from the results list, but also when other users create new resources that are of his interests into the tracking repository.

Compared to the real time delivery mode, the asynchronous delivery happens when the user who may be interested in the new created document is not connected to any application. E-mail and syndication can be used to asynchronously recommend documents to users; at this time, only the e-mail service has been implemented.

After a new document instance is created within the CAM repository (e.g. when a user consults a document), the recommendation service searches for the users interested in the new document, as well as the applications where these users are logged in. If no applications are found, the service builds a mail containing the list of recommended documents and sends it to each of the concerned users.

5. Conclusion and Future Works

This paper presented a solution for recommending documents to students according to their current learning goals and activities. This process is possible thanks to (1) attention metadata gathered when a user performs an action on a
resource, (2) the user and document models described in terms of semantic annotations based on the ACM taxonomy, and (3) an algorithm able to calculate similarities between these models. The recommendation service that implements this algorithm processes synchronous recommendations to online users but also asynchronous recommendations to offline users: links to the recommended documents are integrated into the graphical user interface of the learning environment, whereas emails containing references to the recommended documents are sent to the concerned users. The performance (in term of relevance) of the recommendation algorithm highly depends on the annotated documents collection, but the expertise level of users is of most importance. Indeed, our preliminary tests have shown satisfying results when users are characterized by a high expertise level in the domain covered by the accessed documents (e.g. the concepts defined in the ACM taxonomy).

To consider these limitations, we will work in enhancing the presented recommendation technique by testing multiple algorithmic solutions and mechanisms. The system currently recommends documents to students according to their current activity described in terms of semantic annotations associated to the accessed resources. However, our CAM framework allows collecting not only users’ activities related to resources, but also activities related to applications. Therefore, we plan to use the semantic annotations to describe accessed applications in order to refine the filtering of recommended documents depending on the most suitable applications deploying the documents and matching the user current needs. Also, the recommendation service could implement a machine-learning algorithm: the idea is to make recommendation decisions evolve according to user’s activities, but also to patterns that should be recognized and built based on CAM data. Indeed, lot of information is already available into the CAM repository, and additional attention metadata can easily be defined and collected by agents or sensors. Therefore, our algorithm should integrate others attention metadata as input data; the main work here consists in identifying and calculating these data.

Finally, the semantic-oriented character of the recommendation system enables to develop in the future multiple personalized functionalities such as customized search or browsing facilities. The techniques mentioned above (annotated applications, machine-learning algorithm, additional input CAM data) even offer the opportunity of building entire personalized learning application hosting personalized documents according to the current learner needs.

**References**


