A Personalized Collaborative Digital Library Environment

M. Elena Renda and Umberto Straccia

I.S.T.I. – C.N.R.
Via G. Moruzzi,1 I-56124 Pisa (PI) ITALY
{renda,straccia}@iei.pi.cnr.it

Abstract. We envisage a Digital Library not only as an information resource where users may submit queries to satisfy their information need, but also as a collaborative working and meeting space. We will present a personalized collaborative Digital Library environment, where users may organise the information space according to their own subjective view, may become aware of each other, exchange information and knowledge with each other, may build communities and may get recommendations based on preference patterns of the users.

1 Introduction

It is widely recognised that the internet is growing rapidly in terms of the number of users accessing it, the amount of *Digital Libraries* (DLs) created and accessible through it, and the number of times users use them in order to satisfy their information needs. This has made it increasingly difficult for individuals to control and effectively seek for information among the potentially infinite number of DLs available on the internet.

Typically, DLs provide a search service to the web community at large. A common characteristic of most of these retrieval services is that *they do not provide any personalized support to individual users*, or poorly support them. Indeed, they are oriented towards a generic user, as they answer queries crudely rather than, learn the long-term requirements of a specific user. In practice, users use the same information resource over and over and would benefit from customization: the time consuming effort that the user put in searching documents and possibly downloading them from the DL is often forgotten and lost. Later, the user may wish to perform a search about the same topic to find relevant documents that have, *e.g.* appeared since the last time a search was performed.

This requires a repetition of the manual labour in searching and browsing to find the documents just like the first time. Additionally, users are highly interested in being able to organize the information space according to their own subjective perspective (see *e.g.* [8, 13]). The requirement of personalized search in the context of DLs is already known and some DLs provide related functionality (see *e.g.* [4, 8, 11, 13, 14, 17, 19, 21]). Many of them fall in the category of *alerting services*, *i.e.* services that notify a user (by sending an e-mail), with a list of references to new documents deemed as relevant. Typically all these services are based on the so-called notion of *user profile* (a machine representation of the user's information need). It can be acquired either automatically (by user-system interaction) or set-up manually (by the user). The acquisition of a user profile and the successive matching of documents against it, in order to *filter out* the relevant ones, is known as *Information* or *Content-based Filtering* [2, 12].

Very seldom, except *e.g.* [8], DLs can also be considered as *collaborative meeting* or *working places*, where users may become aware of each other, open communication channels, and exchange information and knowledge with each other or with experts. Indeed, usually users access a DL in search of some information. This means that it is quite probable that users may have overlapping interests if the information available in a DL matches their expectations, backgrounds, or motivations. Such users might well profit from each other's knowledge by sharing opinions or experiences or offering advice. Some users might enter into long-term relationships and eventually evolve into a community if only they were to become aware of each other. Such a service might be important for a DL as it supplies very focussed information. Concerning the information seek task, the *recommendation* of items based on preference patterns of others users is probably the most important one. The use of opinions and knowledge of other users to predict the relevance value of items to be recommended to each user in a community is known as *Collaborative* or *Social Filtering* [3, 5, 15, 16, 20]. These methods are built on the assumption that a good way to find interesting content is to find other users who have similar interests, and then recommend items that those similar users like. In contrast to information filtering methods, collaborative filtering methods do not require any content analysis as they are based on aggregated user ratings of these items.

Both approaches share the common goal of assisting in the users' search for items of interest, and thus attempt to address one of the key research problems of the information age: locating relevant information in a haystack that is growing rapidly. Providing personalized information organisation and search in the context of a collaborative DL environment as additional services to the uniform and generic information search offered today, is likely to be an important step to make relevant information available to people with minimal user effort [1].

The contribution of our paper is as follows: we will (*i*) formalise an abstract collaborative DL environment, where users and communities may search, share and organize their information space according to their own view; (*ii*) present an instance of the environment as the system currently being under development within the EU funded project CYCLADES[1]; and (*iii*) for completeness, we will sketch out the recommendation algorithms. The underlying techniques used for recommendation fall in the afore mentioned filtering methods.

The outline of the paper is as follows. In the next section we will formalise the main concepts of our personalized collaborative DL environment. In Section 3 we will present CYCLADES, while in Section 4 the recommendation methods will be presented briefly. Finally, Section 5 concludes.

[1] www.ercim.org/cyclades

2 A personalized collaborative DL environment

Our personalized collaborative DL environment is made out by several concepts: *actors*, *objects* and *functionality*. Actors will be able to act on objects by means of the DL's functionality. At first, we will give a brief overview of the environment we have in mind and then move on to its formalisation. Roughly, our collaborative environment is based on the *folder paradigm*, *i.e.* users and communities of users may organise the information space into their own folder hierarchy, as may be done with directories in operating systems, bookmark folders in Web browser and folders in e-mail programs. The idea of organising the information space into folders is not new within DLs. For instance, in [8] users are allowed to define folders of bookmarks (*i.e.* URLs). A folder becomes a holder of information items, which are usually semantically related and, thus, implicitly determines what the folder's topic is about. Therefore, rather than to consider user profiles, we will deal with *folder profiles*, *i.e.* representations of what folders are *about*: the user's set of folder profiles represents the set of topics she is interested in.

2.1 Actors

We will distinguish two types of *actors*: the set $\mathcal{U}$ of *users* u and the set $\mathcal{C}$ of *communities* C. A community may be seen as a set of users sharing a common (scientific, professional) background or view of the world. In particular, communities are characterised by a shared interest in the information made available. We postulate that a community $C \in \mathcal{C}$ has a membership function $\mu_C{:}\mathcal{U} \rightarrow \{0,1\}$, where $\mu_C(u) = 1$ (for ease $u \in C$) indicates that the user u belongs to the community C. We do not require that a user has to belong to at least one community, *i.e.* we assume that it is a user's choice to join a community or to leave it. A user may belong to different communities as well. It is not our purpose to address the issue of how a community may be created and which are the policies to join and to leave it. We simply assume that there is a *community administrator* (a user $u^C \in \mathcal{U}$) for each community $C \in \mathcal{C}$, who is in charge of defining these policies (similarly, we will not address the issue of becoming a community administrator within the environment).

2.2 Objects

We will consider three types of *objects*, which may be managed within the environment by users and communities: data items, collections and folders. The objects are organised according to a multilevel model (see Figure 1).

Data Items. At the lowest level, we have the set $\mathcal{D}$ of *data items* d. $\mathcal{D}$ is the information space and the data items are the information resources that a user is usually interested in discovering or searching for within the DL. The data items may be *e.g.* papers, reports, journals, proceedings, notes, annotations, discussions, URIs. A data item might also be just a metadata record, which consists of a set of attributes and related values specifying features of a document, according to a specific schema, *e.g.* Dublin Core [9]. The set of data items $\mathcal{D}$ might well be distributed, heterogeneous in content, format and media (video, audio).

Collections. At the next higher level, we allow the data items $d \in \mathcal{D}$ be grouped into *collections*. A collection may be seen as a set of data items, which are grouped together according to some relatedness criteria, *e.g.* the set of data items created within the same year, or those created by the same author, or those about the same topic, say "collaborative digital libraries", or, more obvious, the set of data items belonging to the same digital archive. We assume that there is a set $\mathcal{L}$ of collections L and a membership function $\mu_L: \mathcal{D} \rightarrow \{0, 1\}$, where $\mu_L(d) = 1$ (for ease $d \in L$) indicates that the data item d belongs to the collection L. We also assume that there is at least one collection in $\mathcal{L}$, called *universal collection* and denoted $L_\top$, which includes all the data items $d \in \mathcal{D}$. Note that a data item may belong to several collections. Furthermore, we do not specify whether the collections are materialised or are just "views" over $\mathcal{D}$. This does not play a significant role in our context. Finally, like for communities, we will assume that for each collection $L \in \mathcal{L}$ there is a *collection administrator* (a user $u^L \in \mathcal{U}$), who is in charge of defining both the collection L and the access policies to it.

Folders. At the third level, we have *folders*. A folder is a container for data items. A folder should be seen as the main environment in which users will carry out their work. Folders may be organised by users according to their own folder hierarchy, *i.e.* a set of hierarchically organised folders, each of which is a repository of the user's selected data items. Each folder typically corresponds to one subject (or discipline, or field) the user is interested in, so that it may be viewed as a thematic repository of data items. In order to accomplish a truly personalized interaction between user and system, this correspondence is implemented in a way which is fully idiosyncratic to the user; this means that *e.g.* a folder named `Knowledge Representation and Reasoning` and owned by user `Tim` will not correspond to any "objective" definition or characterisation of what "knowledge representation and reasoning" is, but will correspond *to what Tim means by* "knowledge representation and reasoning", *i.e.* to his personal view of (or interest in) "knowledge representation and reasoning". As we will see later on, this user-oriented view of folders is realised by learning the "semantics of folders" from the current contents of the folders themselves. We will allow two types of folders: (*i*) *private folders*, *i.e.* a folder owned by a user only. This kind of folder can only be accessed and manipulated by its owner. For other users, they are invisible; and (*ii*) *community folders*, which can be accessed and manipulated by all members of the community who owns the folder. Community folders are used to share data items with other users and to build up a common folder hierarchy. Community folders may also contain *discussion forums* (a kind of data item) where notes may be exchanged in threaded discussions (similar to news groups). Formally, we assume that there is a set $\mathcal{F}$ of (either private or community) folders F. For each user u, with $\langle \mathcal{F}^u, \preceq^u \rangle$, we indicate the user's folder hierarchy, where $\mathcal{F}^u \subseteq \mathcal{F}$, $\preceq^u$ is a tree-like order on $\mathcal{F}^u$ and with $F^u_\top$ we indicate its *home folder* or *top folder*, *i.e.* the root folder of the hierarchy $\langle \mathcal{F}^u, \preceq^u \rangle$. Furthermore, given a folder $F \in \mathcal{F}$, we assume that (*i*) there is a membership function $\mu_F: \mathcal{U} \rightarrow \{0, 1\}$, where $\mu_F(u) = 1$ (for ease $F \in u$) indicates that the folder F belongs to the user's u folder hierarchy, *i.e.* $F \in \mathcal{F}^u$; (*ii*) there is a membership function $\mu_F: \mathcal{C} \to \{0,1\}$, where $\mu_C(d) = 1$ (for ease $F \in C$) indicates that the folder F is a community folder and belongs to the community C; and (*iii*) there is a membership function $\mu_F: \mathcal{D} \to \{0,1\}$, where $\mu_F(d) = 1$ (for ease $d \in F$) indicates that the data item d belongs to the folder F. Figure 1 shows an example of community, users and object organisation. In it, users u_1 and u_2 belong to the same community C_1. User u_2 has no private folders, while F_4 and F_5 belong to the same community C_1.

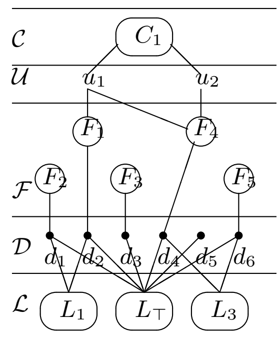

Fig. 1. Personalized information space organisation.

2.3 Actions

A user may perform a set of actions (see below), depending whether she is a member of a community or not, and whether she is a collection or a community administrator. At any time, the user performs her actions with respect to (w.r.t.) the *current folder*. At the beginning this is the user's home folder.

Folder management. A user can perform basic management actions on the folders she has access to: (*ii*) w.r.t. "folder hierarchy", folder management operations include creating a new folder as a child of an existing folder, deleting, moving a folder from an existing parent folder to a new parent folder (community administrators are allowed to manage the folder hierarchy of a community); and (*ii*) w.r.t. "folder content", folder management actions include saving data items from a search session in folders (see below), deleting, undeleting and destroying data items, moving and copying data items from one folder to another, rating, annotating, downloading and uploading data items.

Collection management. A collection administrator can create, edit, delete and define the access policies of collections. New collections may be defined in terms of others, *e.g.* using meet, join and refinement operators.

Collaborative support. Collaboration between users is supported through the possibility of sharing community folders along with their contents and folder structure. Discussion forums may be created within folders to allow informal exchange of notes and arguments. Rating and annotation of data items also may take the form of discussions among the members of a community. In order not to loose shared activity in the collaborative DL environment, mutual awareness may be supported through event icons (a kind of data item) displayed in the environment. Activity reports that are daily received by email may also be possible. Also, users may view the list of all existing communities so that they become aware of ongoing community activity. This does not mean that they can look inside communities, but only *e.g.* the title, the description and the identity of the community administrator are available. To become a member, users may directly join the community if this is allowed by the community's policy, or may contact the administrator to be invited to the community. In summary, collaboration support concerns with inviting or removing members to or from a community, leaving, viewing and joining a community (only for communities open to subscription), contacting community managers or other users (*e.g.* via email), creating discussion forums, adding notes to a discussion forum, editing event notification preferences (icons, daily report) and rating data items.

Search data items. The user can issue a query q, whose result is a partial order (the result list) on the data items $d \in \mathcal{D}$. The user is allowed to store selected data items of the result list within her folder hierarchy. In *ad-hoc search* a user u specifies a query q and the action of the system will be to look for relevant data items within a set of user specified folders $F_i \in \mathcal{F}^u$ she have access to, *i.e.* to search within $\{d \in \mathcal{D} \colon d \in F_i\}$, or to search within a specified collection C, *i.e.* $\{d \in \mathcal{D} \colon d \in C\}$ (we do not specify the syntax of queries, which depends on the indexing capabilities of the underlying DL). We further allow a kind of *filtered search*: this is like to the usual ad-hoc search, except that the user u specifies a query q and a folder $F \in u$, and the action of the system will be to look for data items $d \in \mathcal{D}$ such that d is relevant both to the query and to the folder F. For both types of search there exists widely known methods. Ad-hoc search is the usual task of information retrieval (see [22]), while filtered search may be accomplished in at least two ways: (*i*) through techniques of query expansions [7], *i.e.* we expand the query q with significant terms of the folder profile f of F and then submit the expanded query; or (*ii*) we first issue the query q as an ad-hoc query, and then filter the result list w.r.t. the folder profile [2, 6, 12, 18].

Recommendation. A user may get recommendations of *data items*, *collections*, *users*, and *communities* issued to users based on other users' (implicit or explicit) ratings, and on the perceived similarity between the interests of the user, as represented by a given folder, and the interests of these other users, as represented by their folders. All recommendations are specific to a given user folder, *i.e.* they have always to be understood in the context not of the general interests of the user, but of the specific interests (topic) of the user represented by a folder.

Without doubt, the above set of actions provides us an enhanced personalized collaborative DL environment. Several of the items above are eligible to be the subject of deeper investigations but, we will not address them further.

3 An application: CYCLADES

The model of a personalized collaborative DL environment we have presented, is currently under implementation in the CYCLADES system. The main goal of CYCLADES is the development of a system, which provides an open collaborative virtual archive environment, which (among others) supports users, communities (and their members) with functionality for (*i*) advanced search in *large, heterogeneous, multidisciplinary digital archives* (*ii*) collaboration; and (*iii*) filtering and recommendation. With respect to the model described in Section 2, a main feature of CYCLADES is that it will use the protocol specified by the Open Archives Initiative[2] (OAI) to harvest and index metadata records from any archive that supports the OAI standard. As a consequence, the set $\mathcal{D}$ of data items includes the set of metadata records harvested from the OAI compliant archives. As a reminder, the OAI is an agreement between several Digital Archives in order to provide interoperability. The specifications give *data providers* (individual archives) easy-to-implement mechanisms for making the documents' metadata records in their archives externally available. This external availability then makes it possible for *service providers* to build higher levels of functionality. The CYCLADES system is indeed such a service provider. From a logical point of view we may depict the functionality of the CYCLADES system as in Figure 2, which highlights the functionality related to *collaboration, search, filtering and recommendation* of data items grouped into collections. Figure 3 shows a mock-up of the user interface, while Figure 4 shows its architecture.

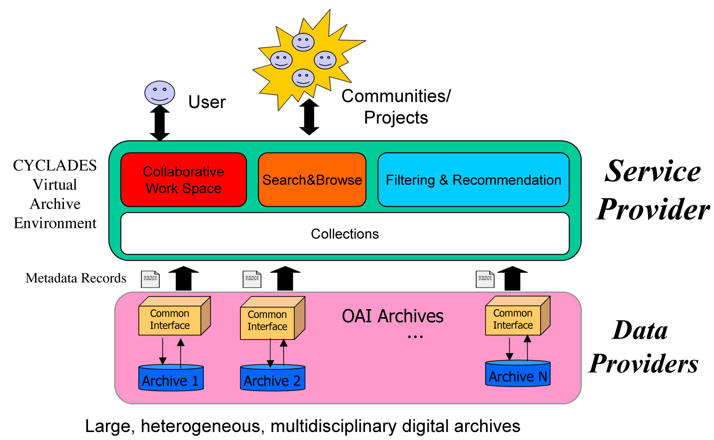

Fig. 2. Logical view of CYCLADES functionality.

It should be noted that from an architecture point of view each box is a Web service distributed over the internet. The CYCLADES system, which will be accessible through Web browsers, provides the user with different environments, according to the actions the user wants to perform.

The *Collaborative Work Service* provides the folder-based environment for managing metadata records, queries, collections, external documents, ratings and annotations. Furthermore, it supports collaboration between CYCLADES users by way of folder sharing in communities.

The *Search and Browse Service* supports the activity of searching records from the various collections, formulating and reusing queries, and browsing schemas, attribute values, and metadata records.

[2] www.openarchives.org

The *Access Service* is in charge of interfacing with the underlying metadata archives. In this project, only archives adhering to the OAI specification will be accounted for; however, the system is extensible to other kinds of archives by modifying the Access Service only.

The *Collection Service* manages collections (*i.e.* their definition, creation, and update), thus allowing a partitioning of the information space according to the users' interests, and making the individual archives transparent to the user.

The *Filtering and Recommendation Service* provides filtered search, recommendations of records, collections, users, and communities.

The *Mediator Service*, the entry point to the CYCLADES system, acts as a registry for the other services, checks if a user is entitled to use the system, and ensures that the other services are only called after proper authentication. All of these services interoperate in a distributed environment. Security and

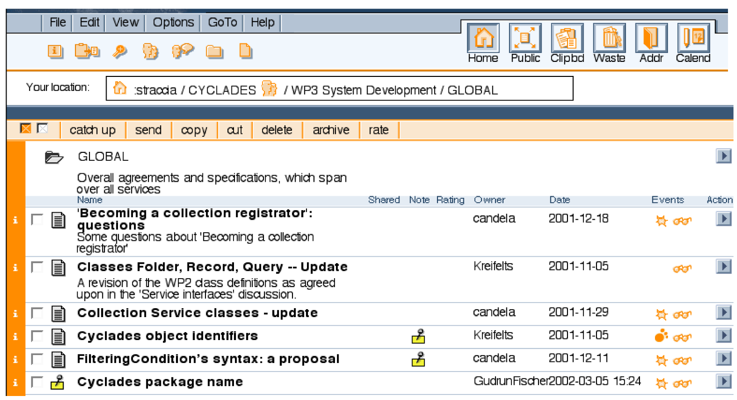

Fig. 3. User interface (mock-up).

system administration will be provided for centrally (by the Mediator Service). The CYCLADES services can run on different machines, and will only need a HTTP connection to communicate and collaborate.

4 Recommendation algorithms

A consequence of our environment is that, (*i*) by allowing users to organise the information space according to their own subjective view; and (*ii*) by supporting a collaborative environment, it is possible to provide a set of recommendation functionality that, to the best of our knowledge, have not yet been investigated. Indeed, the recommendations regard not only the data items and the collections made available by the DL, but also the users and communities. Due to space limitation, we will just sketch out the algorithms. The algorithms below are those implemented in the CYCLADES system.

Preliminaries. For ease of presentation, we will assume that data items are pieces of text (*e.g.* text documents). It is worth noting that our algorithms can be extended to manage data items of different media kind, like audio and video. By t_k, d_j, and F_i we will denote a text term, a data item, and a folder, respectively. Terms are usually identified either with the words, or with the stems of words, occurring in data items. For ease, following the well-known

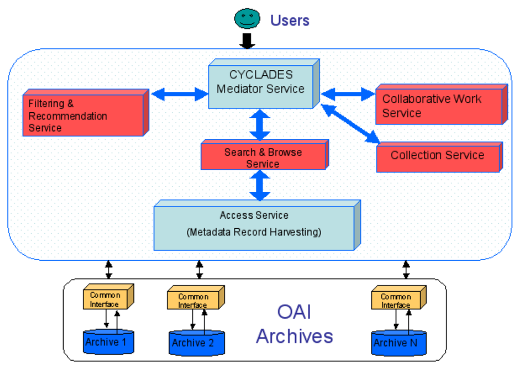

Fig. 4. Architecture.

vector space model [22], a data item d_j is represented as a vector of *weights* $d_j = \langle w_{j1}, \ldots, w_{jm} \rangle$, where $0 \leq w_{jk} \leq 1$ corresponds to the "importance value" that term t_k has in the data item d_j, and m is the total number of unique terms in the indexed universal collection $L_\top$. The *folder profile* (denoted f_i) for folder F_i is computed as the *centroid* of the data items belonging to F_i. This means that the profile of F_i may be seen as a data item itself [2] (*i.e.* the mean, or prototypical, data item of F_i) and, thus, it is represented as a vector of weighted terms as well, *i.e.* $f_i = \langle w_{i1}, \ldots, w_{im} \rangle$. Of course, more complicated approaches for determining the folder profile may be considered as well, *e.g.* taking into account the hierarchical structure of the folders (see, *e.g.* [10]). Conceptually, they do not change much in our algorithm. Given a folder F_i, a data item $d_j \in F_i$ and a user $u_k \in \mathcal{U}$ such that $F_i \in u_k$, by $0 \leq r_{ijk} \leq 1$ we denote the *rating* given by user u_k to data item d_j relative to folder F_i (a data item within a community folder, may be accessed, *e.g.* read, annotated and rated, by many different users). We further assume that whenever a data item d_j belongs to a folder F_i of a user u_k, an *implicit* default rating $\breve{r}$ is assigned. Indeed, the fact that $d_j \in F_i \in \mathcal{F}^{u_k}$ is an implicit indicator of being d_j relevant to folder F_i for user u_k. Finally, we average out the ratings r_{ijk} given by users u_k relative to the same data item–folder pair (i, j) and indicate it as r_{ij}.

In summary, we may represent (i) the data items as a 2-dimensional matrix, where a row represents a data item d_j and a column represents a term t_k. The value of the cell is the weight w_{jk} of term t_k in the data item d_j; (ii) the folder profiles as a 2-dimensional matrix, where a row represents a folder profile f_i and a column represents a term t_k. The value of the cell is the weight w_{ik} of term t_k in the folder profile f_i; and (iii) the ratings as a 2-dimensional matrix, where a row represents a folder F_i and a column represents a data item d_j. The value of the cell is the rating r_{ij}. The three matrixes are shown in Table 1, where $v = |\mathcal{F}|$ is the number of folders and $n = |L_\top|$ in the number of data items.

The *content similarity* (denoted $CSim(\cdot,\cdot)$) between two data items, or between a data item and a folder profile, or between two folder profiles is a correlation coefficient (*e.g.* *cosine*) among two rows within the matrixes (a) and (b) of Table 1. Similarly, the *rating similarity* of two folders F_1 and F_2 (denoted $RSim(F_1, F_2)$) can be determined as a correlation [5, 16] (*e.g.* *Pearson correlation coefficient*) between two rows of the matrix (c) in Table 1. Finally, the *similarity* (denoted $Sim(F_1, F_2)$ between two folders F_1 and F_2, which takes into account both the content and collaborative aspects, can be determined as a linear combination between their content similarity and their rating similarity.

Our recommendation algorithms follow a four-step schema described below. Let u be a user and let $F \in u$ be a folder (the *target folder*) for which the recommended items should be found. The algorithm schema is as follows: (i) select the set of most similar folders F_i to F, according to the similarity measure Sim; (ii) from this set, determine a pool of possible recommendable items; (iii) for each of the items in the pool compute a recommendation score; (iv) select and recommend a subset of items with highest score, and not yet recommended to F. We proceed now with a more detailed description of the above algorithm, specialised for the two cases of user recommendation[3] and of data items.

Recommendation of users. (i) Select the set $MS(F)$ of most similar folders to the target folder $F \in u$; (ii) for each folder $F_i \in MS(F)$, consider the users for which the folder F_i belongs to their folder hierarchy, *i.e.* compute the *pool of possible recommendable users* $P_U = \{u' \in \mathcal{U} \colon \exists F_i . F_i \in MS(F), F_i \in u'\} \setminus \{u\}$; ($iii$) compute the recommendation score for each possible recommendable user, *i.e.* for each user $u' \in P_U$ determine the *user hits factor* $h(u') = |\{F_i : F_i \in MS(F), F_i \in u'\}|$ (the number of folders F_i judged as similar to the target folder F belonging to user u'). For each user $u' \in P_U$ the *recommendation score* $s(F, u')$ is computed as follow: $s(F, u') = h(u') \cdot \sum_{F_i \in MS(F), F_i \in u'} Sim(F, F_i)$; and ($iv$) according to the recommendation score, select a set of most recommendable users, not yet recommended to the target folder F.

Note that the more a folder $F_i \in u'$ is similar to the target folder $F \in u$, the more related, in terms of interests, are the users u' and u. Additionally, the more similar folders belong to the same user u', the more this u''s interests overlap those of user u, which explains the computation of the recommendation score.

Recommendation of data items. The only difference with the previous one concerns the computation of the recommendable data items and their recommendation score. Indeed, we will exploit the fact that data items are pieces of text and that there might be ratings associated: (i) the *pool of possible recommendable data items* is the set of data items belonging to the folders $F_i \in MS(F)$, *i.e.* $P_D = \{d \in \mathcal{D} \colon \exists F_i . F_i \in MS(F), d \in F_i\} \setminus \{d \in \mathcal{D} \colon \exists F' \in u, d \in F'\}$ (we do not recommend data items already known to the user); (ii) the recommendation score for $d_j \in P_D$ w.r.t. F is computed as a linear combination of the *content-based* and the *rating-based recommendation scores*. The content-based recommendation score of $d_j \in P_D$ w.r.t. the target folder F is the content similarity between d_j and the folder profile of F. The ratings-based recommendation score

[3] The recommendation of communities and collections are quite similar.

	t_1	...	t_k	...	t_m
d_1	w_{11}	...	w_{1k}	...	w_{1m}
d_2	w_{21}	...	w_{2k}	...	w_{2m}
...	...	...	...	...	...
d_j	w_{j1}	...	w_{jk}	...	w_{jm}
...	...	...	...	...	...
d_n	r_{n1}	...	w_{nk}	...	w_{nm}

(*a*)

	t_1	...	t_k	...	t_m
f_1	w_{11}	...	w_{1k}	...	w_{1m}
f_2	w_{21}	...	w_{2k}	...	w_{2m}
...	...	...	...	...	...
f_i	w_{i1}	...	w_{ik}	...	w_{im}
...	...	...	...	...	...
f_v	w_{v1}	...	w_{vk}	...	w_{vm}

(*b*)

	d_1	...	d_j	...	d_n
F_1	r_{11}	...	r_{1j}	...	r_{1n}
F_2	r_{21}	...	r_{2j}	...	r_{2n}
...	...	...	...	...	...
F_i	r_{i1}	...	r_{ij}	...	r_{in}
...	...	...	...	...	...
F_v	r_{v1}	...	r_{vj}	...	r_{vn}

(*c*)

Table 1. (*a*) The data item matrix. (*b*) The folder profile matrix. (*c*) The folder-data item rating matrix.

of d_j w.r.t. F is the weighted sum $s^R(F, d_j) = \overline{r} + \frac{\sum_{F_i \in MS(F)} (r_{ij} - \overline{r}_i) \cdot RSim(f, f_i)}{\sum_{F_i \in MS(F)} \cdot RSim(f, f_i)}$, where $\overline{r}$ ($\overline{r}_i$) is the mean of the ratings in the target folder F.

5 Conclusions

We envisage a Digital Library not only as an information resource where users may submit queries to satisfy their information need, but also as a collaborative working and meeting space. Indeed, users looking within an information resource for relevant data might have overlapping interests, which may turn out to be of reciprocal interest for the users: users might well profit from each other's knowledge by sharing opinions and experiences. As such, we have formalised a personalized collaborative Digital Library environment in which the user functionality may be organised into four categories: users may (*i*) search for information; (*ii*) organise the information space (according to the "folder paradigm"); (*iii*) collaborate with other users sharing similar interests; and (*iv*) get recommendations. We also described the CYCLADES system, which is indeed an on going implementation of the environment. We are aware that many concepts and techniques presented in this paper are eligible to be the subject of further investigations, which we will address in the future.

Acknowledgements

This work is funded by the European Community in the context of the CYCLADES project IST-2000-25456, under the Information Societies Technology programme.

References

1. G. Amato and U. Straccia. User profile and applications to digital libraries. In *Proc. 3rd European Conf. on Research and Advanced Technology for Digital Libraries (ECDL-99)*, LNCS 1696, pages 184–197, Paris, France, 1999. Springer-Verlag.
2. N. J. Belkin and B. W. Croft. Information filtering and information retrieval: Two sides of the same coin? *Comm. of the ACM*, 35(12):29–38, 1992.

3. D. Billsus and M. J. Pazzani. Learning collaborative information filters. In *Proc. 15th International Conf. on Machine Learning*, pages 46–54. Morgan Kaufmann, San Francisco, CA, 1998.
4. K. Bollacker, S. Lawrence, and C. L. Giles. A system for automatic personalized tracking of scientific literature on the web. In *DL 99 - The 4th ACM Conf. on Digital Libraries*, pages 105–113, New York, 1999. ACM Press.
5. J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. 14th Annual Conf. on Uncertainty in Artificial Intelligence (UAI-98)*, pages 43–52, Madison, Wisconsin, USA, 1998.
6. J. Callan. Learning while filtering documents. In *Proc of the 21th Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval (ACM SIGIR-98)*, pages 224–231, Melbourne, Australia, 1998.
7. C. Carpineto, R. De Mori, G. Romano, and B. Bigi. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems*, 19(1):1–27, 2001.
8. M. Di Giacomo, D. Mahoney, J. Bollen, A. Monroy-Hernandez, and C. M. Rouiz Meraz. Mylibrary, a personalization service for digital library environments, 2001.
9. DublinCore. Dublin core metadata element set. `purl.org/metadata/dublin_core`, WWW.
10. S. Dumais and H. Chen. Hierarchical classification of web content. In *Proc. 23rd Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR-00)*, pages 256–263, Athens, Greece, 2000.
11. D. Faensen, L. Faulstich, H. Schweppe, A. Hinze, and A. Steidinger. Hermes: a notification service for digital libraries. In *ACM/IEEE Joint Conference on Digital Libraries*, pages 373–380, 2001.
12. C. Faloutsos and D. W. Oard. A survey of information retrieval and filtering methods. University of Maryland Technical Report CS-TR-3514, 1995.
13. L. Fernandez, J. A. Sanchez, and A. Garcia. Mibiblio: personal spaces in a digital library universe. In *ACM DL*, pages 232–233, 2000.
14. P. W. Foltz and S. T. Dumais. Personalized information delivery: An analysis of information filtering methods. *Comm. of the ACM*, 35(12):51–60, 1992.
15. D. J. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave information tapestry. *Comm. of the ACM*, 35(12):61–70, 1992.
16. J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proc. 22th Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR-99)*, pages 230–237, Berkeley, CA USA, 1999.
17. *Information Filtering Resources*. `www.enee.umd.edu/medlab/filter/filter.html`.
18. J. Mostafa, S. Mukhopadhyay, W. Lam, and M. Palakal. A multilevel approach to intelligent information filtering: Model, system, and evaluation. *ACM Transactions on Information Systems*, 15(4):368–399, 1997.
19. A. Moukas. *Amalthaea*: Information discovery and filtering using a multiagent evolving ecosystem. In *Proc. Practical Applications of Agents and Multiagent Technology*, London, GB, 1996.
20. P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proc. ACM 1994 Conf. on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.
21. L. M. Rocha. Talkmine and the adaptive recommendation project. In *ACM DL*, pages 242–243, 1999.
22. G. Salton and J. M. McGill. *Introduction to Modern Information Retrieval*. Addison Wesley Publ. Co., Reading, Massachussetts, 1983.