Semantically Rich Recommendations in Social Networks for Sharing and Exchanging Semantic Context

Stefania Ghita, Wolfgang Nejdl, and Raluca Paiu

L3S Research Center / University of Hanover
Deutscher Pavillon, Expo Plaza 1
30539 Hanover, Germany
{ghita,nejdl,paiu}@l3s.de

Abstract. Recommender algorithms have been quite successfully employed in a variety of scenarios from filtering applications to recommendations of movies and books at Amazon.com. However, all these algorithms focus on single item recommendations and do not consider any more complex recommendation structures. This paper explores how semantically rich complex recommendation structures, represented as RDF graphs, can be exchanged and shared in a distributed social network. After presenting a motivating scenario we define several annotation ontologies we use in order to describe context information on the user desktop and show how search result ranking can be built on this information. We then discuss how social distributed networks and interest groups are specified using extended FOAF vocabulary, and how members of such interest groups share semantically rich recommendations in such a network. These recommendations transport shared context information as well as ranking information, as described in the annotation ontologies.

1 Introduction

This paper explores how we can use communication in social networks to share and extend context information and how semantically rich recommendations within members of common interest groups in such settings can be realized. We will build upon FOAF networks, which describe personal and group information, based on the FOAF vocabulary to describe friends, groups and interests. We will focus on how to share context in such a network, how to use these shared metadata to connect the information of different peers in the social network and how to use it for social recommendations.

The next section describes a motivating scenario that shows how context and rankings are exchanged inside a research group. Section 3 discusses how to describe contexts and their corresponding metadata by means of appropriate ontologies and how to use these metadata for extended desktop search with appropriate ranking of search results. In section 4.1 we present how we extend the FOAF vocabulary in order to specify interest groups and shared information. Section 4 describes how we actually exchange context and importance information among the members of an interest group. Section 5 gives an overview of related work, we conclude by sketching some future research issues.

2 Motivating Scenario

As our motivating scenario, let us consider our L3S Research Group context and within this group, Bob and Alice as two members who exchange information. One important task in a research group is exchanging and sharing knowledge, which we will focus upon in this paper. Unfortunately, the most widely used infrastructure for this purpose, email, is poorly suited to support this exchange. When we exchange documents by email, no context is shared (for example which are the interesting follow-up papers, or which are the interesting references for a paper) and any comments about the documents that are included in the email are basically lost as soon as the attached documents are stored in some directory.

The following example shows how such a sharing scenario can be supported in a more efficient manner. In our example we assume that Bob mails Alice a document which he sent to the DELOS Workshop, with the title "I know I stored it somewhere - Contextual Information and Ranking on Our Desktop". Bob is one of the authors and therefore he already has all the important context for this paper including the cited papers stored on his computer. In this first email, Alice will therefore not only receive the paper but also its immediate context relevant for the research group, containing information about all papers that are referenced in the DELOS paper, information about important authors for this topic or which conferences are relevant. In other words, whenever we send a paper, the metadata associated to that paper will also be sent. From the five references included, Alice decides that "ObjectRank: Authority-Based Keyword Search in Databases" and "Activity Based Metadata for Semantic Desktop Search" are of particular interest for her and she sends back an email to Bob requiring additional information about those. As an answer, she receives from Bob the context information associated with these papers, containing the references that Bob has already downloaded. So the context information will be exchanged progressively, from the immediate context to the more distant one.

Figures 1 and 2 present the context created on Alice's desktop as result of her metadata exchange with Bob. Figure 1 contains only the *cites* relationship among the various resources, while in figure 2 we represent additional relationships, like *presented_at*, *downloaded_from*, *author*, or *same_session*. Note that the context networks created on the users' desktop are not separated, but just visualized separately in these figures.

By examining the context graph in figure 2, we see that all the papers labeled from **G** to **O** were presented at different WWW Conferences, in different years, and all were downloaded from the ACM Portal. Papers **A**, **C** and **K** all share the same author, *Bob*, and have been downloaded from the L3S Publication page. Similarly, the publication labeled **B** and the other two papers which were presented at the same session at the VLDB conference, were downloaded from the VLDB web site.

All this information is taken into account when computing the importance of the resources on Alice's desktop. For example, when computing the importance of the conferences, the WWW Conference will be more important than other conferences, since Alice already has a lot of important publications which have been presented there. The number of papers from the same author Alice has already downloaded, also influences how important she considers that author. This means that certain authors are more important than others, based on the publications used and cited in the L3S Research Group,
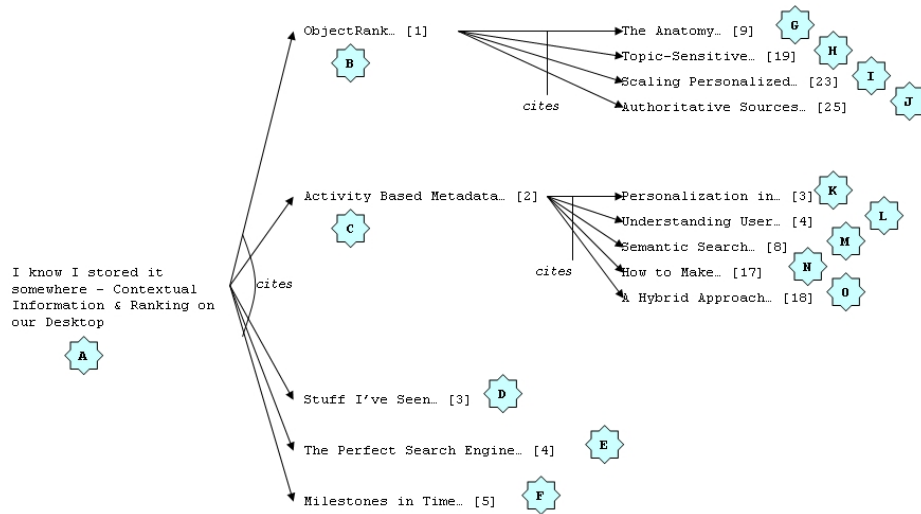
Fig. 1. Publications Context Example - Part 1

as well as on general citation information about these authors. The fact that Alice knows Bob and Bob is one of the authors of three publications Alice has on her desktop influences the importance of Bob's publications and of course, Bob's importance as author. So he will be definitely more important to Alice than other authors not known to her.

In order to be able to compute the rankings of their documents Alice and Bob have to build a context around the resources they have stored on their desktops. The next section presents in more detail how this context information is created and describes how this context is used in computing rankings of search results on the desktop.

3 Representing Context and Importance

3.1 Representing Context

Generally speaking, context information describes all aspects important for a certain situation: ideas, facts, persons, publications, and many more. Context information includes all relevant relationships as well as interaction history. Current desktop search prototypes fall short of utilizing any desktop specific information, especially context information, and just use full text index search. In our scenario we clearly need to use additional context information, and specifically want to exploit the following contexts:

CiteSeer context. The most important aspects we want to record from the CiteSeer context are the publications we are viewing or downloading and how these publications are connected to other publications. Important parts of the available context information are the authors of these publications, the conferences in which they were presented or the year when they were published and even more, the publications which cite them or are cited by them. We want to keep track whether we saved a certain publication on our
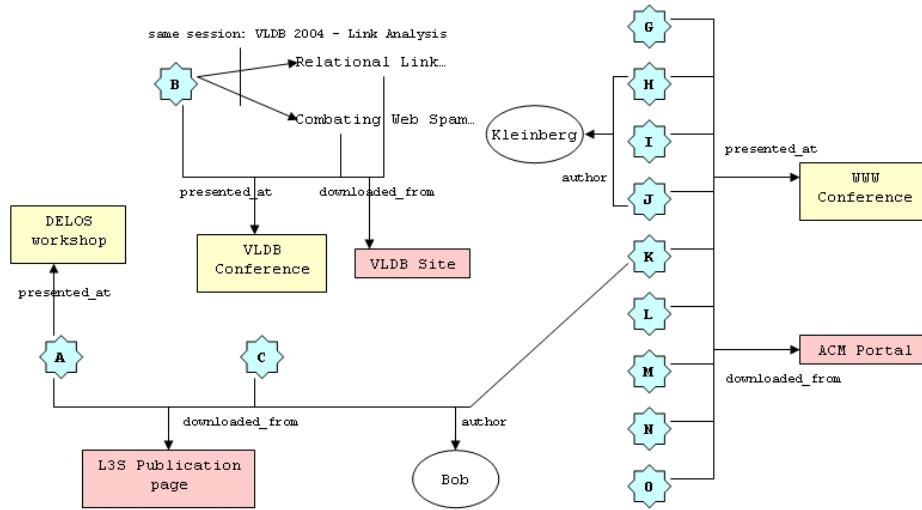
Fig. 2. Publications Context Example - Part 2

own desktop in order to be able to find it later and we want to receive suggestions about papers that might be interesting in the same or overlapping contexts.

CiteSeer provides 4 additional types of links that can be followed after identifying a paper. The most expressive in our case would be the ones that refer to the related documents from co-citations and the papers that appear on the same web site.

Browsing and Desktop context. Browser caches include all information about user's browsing behaviour, which are useful both for finding relevant results, and for providing additional context for results. In our scenario, when we search for a document we downloaded from the CiteSeer repository, we do not only want to retrieve the specific document, but also all the referenced and referring papers which we downloaded on that occasion as well.

In general, we view documents stored from emails and from web sites as our personal digital library, which holds the papers we are interested in, plus all relevant contextual information. When we store documents, we can then retrieve them efficiently and restore the original context we built up when storing these documents. Personalized search and ranking on the desktop takes this contextual information into account as well as the preferences implicit in this information. [6] discusses how people tend to associate things to certain contexts. So far, however, search infrastructures neither collect nor use this contextual information.

Scenario specific annotation ontologies. Figure 3 presents the current prototype ontology, which is used for implementing our motivating scenario and specifies context metadata for the CiteSeer context, files and web pages, together with the relations among them (described in more detail in [2]). Conceptually, the elements in the rectangles rep-

resent classes, circles represent class attributes. We use classes whenever we want to attach importance/ rank on entities, attributes otherwise.
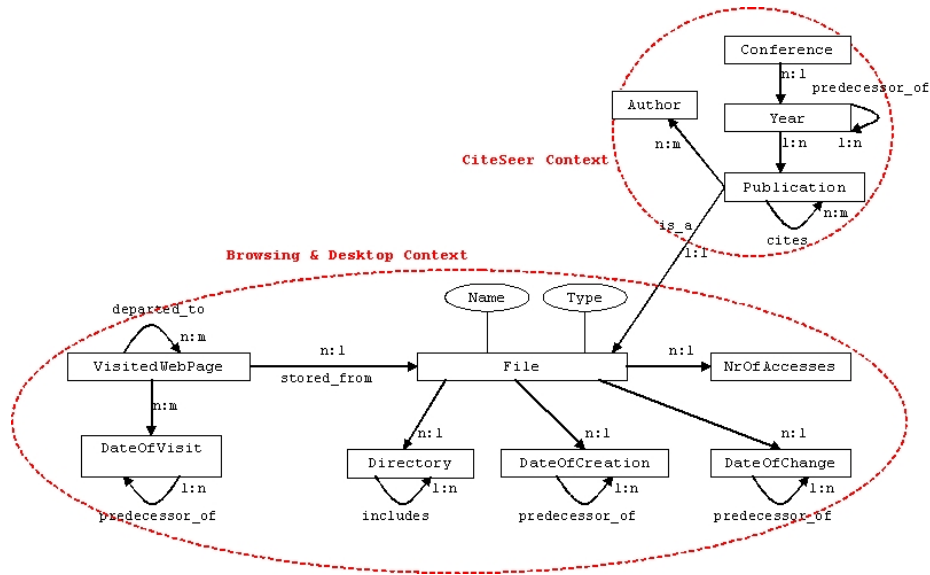

Fig. 3. Context ontology for our prototype

For the browsing and desktop context, we annotate each page with additional information about its basic properties (URL, access date, etc), as well as more complex ones such as in- and out-going links browsed [[2]]. The user's behaviour as the pages or publications he browsed or downloaded provide useful additional information. Files, which are stored from web pages, reside in certain directories, which in turn can include other directories. The creation or change date of a file together with the number of accesses are some other important indicators which have to be taken into account when describing the desktop context. An extended publication ontology makes use of additional knowledge about how CiteSeer pages are connected and what they represent. Publications are referenced by other publications and can cite others, they can have a publication date/ year associated with them, as well as a conference or journal. Publications have authors and are stored as documents on the desktop.

Other ontologies describe contexts like conferences, including reviewers, papers, meetings, authors, or private contexts like birthdays, including persons, locations, etc.

3.2 Representing Importance

In addition to the information which resources are included in a specific context, we also want to know how important or valuable these resources are. We therefore have to develop a mechanism which allows us to express this information and use it for ranking search results.

Authority transfer annotations. Annotation ontologies describe all aspects and relationships among resources which influence the ranking. The identity of the authors, for example, influences our opinion of documents so "author" should be represented explicitly as a class in our publication ontology. We then have to specify how these aspects influence each other's importance.

ObjectRank [1] has introduced the notion of authority transfer schema graphs, which extend schemas similar to the ontologies previously described, by adding weights and edges in order to express how importance propagates among the entities and resources inside the ontology. These weights and edges represent the authority transfer annotations, which extend our context ontologies with the information we need to compute ranks for all instances of the classes defined in the context ontologies.[1]
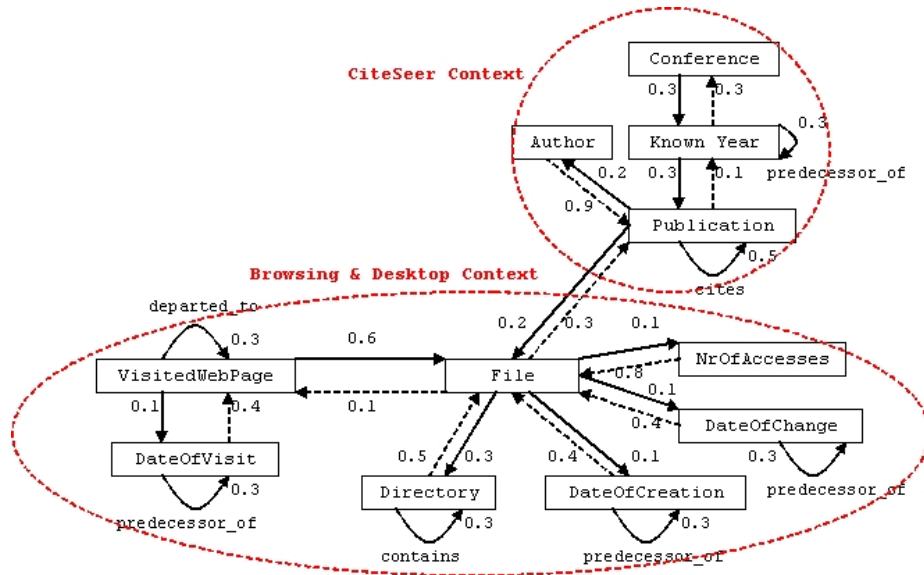
Fig. 4. Authority transfer annotations, including external ranking sources

Figure 4 depicts our context ontology plus its authority transfer annotations. The ontology representing our browsing and desktop context says that a visited web page is important if we arrived at the current one from an important page, if the file under which it is stored is important, or if the date when the page was visited is important. For the CiteSeer context, publications transfer part of their authority to other papers they cite, to their authors, to the files under which they are stored, and to the year when the paper was published. As we can see, citing important papers doesn't make a paper important. As suggested in [1], every edge from the schema graph is split into two edges, one for each direction. This is motivated by the observation that authority potentially flows in

---

[1] In contrast to ObjectRank, we do not compute a keyword-specific ranking, but a global one.

both directions and not only in the direction that appears in the schema (if we know that a particular person is important, we also want to have all emails we receive from this person ranked higher). The final ObjectRank value for each resource is calculated based on the PageRank formula.

Personalized Preferences and Ranking. Different authority transfer weights express different preferences of the user, translating into personalized ranking. The important requirement for doing this successfully is that we include in a user ontology all concepts, which influence our ranking function. For example, if we view a publication important because it was written by an author important to us, we have to represent that in our context ontology. Another example are digital photos, whose importance is usually heavily influenced by the event or the location where they were taken. In this case both event and location have to be included as classes in our context ontology.

4 Sharing Context and Importance

4.1 Interest Groups

Interest groups in our context are specialized social networks that have a stated common interest which connects the members of the group. One important reason for creating interest groups resides in increasing the efficiency of the information flow inside that group. All members of the same interest group share the same domain of interest and the social relationships are woven around this type of information sharing. They are all possibly part of the same professional group, just as we described in the motivating scenario, Alice and Bob being in the same research group, the L3S Research Group.

We chose to represent interest groups based on an extension of FOAF in order to describe the social network of participants and we will describe all contexts as RDF metadata, as presented in [2].

Being based on RDF, FOAF inherits some of its benefits, like the ease of aggregating and harvesting it, or combining it with other vocabularies, thus allowing us to capture a rich set of metadata. The basic FOAF vocabulary itself is pretty simple, pragmatic and designed to allow simultaneous deployment and extension. It is identified by the namespace URI 'http://xmlns.com/foaf/0.1/' and described in more detail at the FOAF project page [11].

FOAF terms represent information which can be grouped in the following five broad categories: FOAF Basics, Personal Information, Online Accounts/ IM, Projects and Groups, Documents and Images. The most important for us is the *Projects and Groups* category, which allows us to talk about groups and group membership among others.

Groups are represented with the aid of the **foaf:Group** class, which represents a collection of individual agents. This concept is intentionally quite broad, covering informal and ad-hoc groups, long-lived communities, organizational groups within a workplace, etc. FOAF provides a mechanism, the **foaf:membershipClass** property, which relates a **foaf:Group** to a sub-class of the class **foaf:Agent** who are members of the group. A similar property is **foaf:member**, which allows us to explicitly express the membership of agents to a group. Persons can also be members of a group, since the **foaf:Person** class is a sub-class of the **foaf:Agent**. This is because all people are considered 'agents'

in FOAF. When describing the members of a group, one can specify their interests by using specific properties, like **foaf:interest**, **foaf:topic_interest**, or **foaf:topic**, even though it is not yet clear how to use them correctly.

A notable omission in the basic FOAF vocabulary is the inability to express anything related to information sharing in a group. Even though being in a group or social network usually means that we want to share information within this social network, there is no vocabulary to express this in FOAF. The assumption we make in this paper is that people belonging to a common interest group will share a specific set of metadata. In our scenario these are the contextual metadata defined by appropriate annotation ontologies, as discussed in the previous section. Thus, when members of an interest group express that they want to share a certain set of metadata, they will agree on an appropriate ontology defining this set. We will therefore extend the FOAF vocabulary with a new property **foaf:shared_context** which takes as its value the annotation ontology describing the metadata to be shared.

Based on this, the FOAF description of the L3S Research interest group and its members Bob and Alice as presented in our motivating scenario looks as follows:

$<foaf:Group>$
  $<foaf:name>L3SResearchGroup</foaf:name>$
  $<foaf:member>$
    $<foaf:Person>$
        $<foaf:name>Alice</foaf:name>$
        $<foaf:homepage\ rdf:resource="http://www.l3s.de/\sim alice"/>$
    $</foaf:Person>$
  $</foaf:member>$
  $<foaf:member>$
    $<foaf:Person>$
      $<foaf:name>Bob</foaf:name>$
      $<foaf:homepage\ rdf:resource="http://www.l3s.de/\sim bob"/>$
    $</foaf:Person>$
  $</foaf:member>$
  $<foaf:shared\_context$
  $rdf:resource=http://www.l3s.de/isearch/citeseerContext.rdf/>$
  $<foaf:shared\_context$
  $rdf:resource=http://www.l3s.de/isearch/browsingDesktopContext.rdf/>$
$</foaf:Group>$

4.2 Exchanging Context Within Interest Groups

Sharing context in an interest group is useful and necessary because not only do we want to publish our own work but we also want to find out about additional new resources related to our work and get suggestions about possible further developments in that area. Recommendation then means suggesting additional related information to given items. In our motivating scenario, we have as interest group a set of researchers, and a set of ontologies defining which metadata are shared between them. The contextual metadata corresponding to those ontologies as discussed in section 3 represent the context information we have available on our desktop.

These context metadata are generated locally by a set of metadata generators [2], which record user actions as well as interactions and information exchanges between members of a group. These metadata generators create RDF annotation files for each resource whose context they describe, so for each relevant resource on the desktop (e.g. a specific publication) we will have this additional RDF information available.

Based on this we are currently implementing a helper application to extend emails client such that for each document attached to an email we can easily add the appropriate context information in RDF format as an additional attachment as well. The recipient of such an email then not only stores the document sent, but the appropriate context information as well. To make this feasible we have extended the contextual metadata described in [2] with additional information about the origin of this information, in this case which user and application generated this metadata.

So in our scenario, whenever Bob sends a publication to Alice, who is member of the same interest group, he will attach the appropriate context information, i.e. the publication context we have discussed in the scenario. The email helper application therefore has to check who is the recipient of an email, which group she belongs to, and therefore which context information/ metadata to attach.

All references in the context metadata use URIs to uniquely identify that resource. For example, if Alice has a file named "bear.wmv", the unique identifier of the file is "file:///C:/Dokumente%20und%20Einstellungen/alice/Desktop/bear.wmv". The default attachment generated by our helper application includes exactly the context described by the ontology to be shared. Additional context information, such as the references for one of the referenced publications, is shared through a sequence of email requests for context information of specific resources identified by the appropriate URIs.

Exchanging context information can be implemented in other ways as well, for example by using a centralized infrastructure, which stores available contextual metadata retrieved through crawling. In contrast to the exchange of contextual information through email attachments, additional measures are necessary to guarantee that only information is shared as specified by the user. Building upon a peer-to-peer infrastructure such as Edutella [8] is another more interesting possibility, extended with appropriate access control capabilities such as described in [4].

Besides these security and privacy issues, only rudimentarily addressed in our current prototype, additional research will have to investigate other models of context exchange beyond the simple iterative sequence of exchange actions described above.

4.3 Sharing Importance

In our distributed scenario, each user has his own contextual network/ context metadata graph and for each node in this network the appropriate ranking as computed by the algorithm described in section 3.2. The computation of rankings on one's desktop is based on the link structure of the resources as specified by the defined ontologies and the corresponding metadata. When sharing information within the group/ network we exchange not only certain contexts but also rankings. So exchanging context information has also an impact on the ranking of results of the desktop search. These values are then recomputed according to the rankings received together with the context from other persons.

Ranking of resources is calculated based on the PageRank formula:

$$r = dAr + (1 - d) \cdot e/|V| \tag{1}$$

applying the random surfer model and including all nodes in the base set. *A* is the adjacency matrix which connects all available instances of the existing context ontology on one's desktop. The weights of the links between the instances correspond to the weights specified in the authority transfer annotation ontology. Thus, when instantiating the authority transfer annotation ontology for the resources existing on the users' desktop, the corresponding matrix *A* will have elements which can be either 0, if there is no edge between the corresponding entities in the data graph, or they have the value of the weight assigned to the edge determined by these entities, in the authority transfer annotation ontology.

When computing the ranking, the matrix, previously defined, has to be transformed into a stochastic matrix. This means that the sum of the elements on every column has to be 1. Additionally we have to take into account the random jump, to an arbitrary entity from the data graph. This aspect is suggested in the following equation:

$$r = d \cdot Normalize(A) + (1 - d) \cdot e/|V| \tag{2}$$

The context which was received from the other members of the interest group is used for building the user's own context, which means that it is also taken into account when creating the adjacency matrix *A*. In order to include the rankings of other users into the computation of the user's own ranking, we work on the vector *e*, which models the random jump. So, if a resource is highly ranked according to the received rankings and the user wants to take this into account, she will simply have to assign a higher value for the corresponding element in the vector which simulates the random jump.

We currently do not take different reputations into account, but clearly if somebody whom I trust and who is important for me sends his recommendations, then I want his suggestions to be higher ranked than the ones received from a more untrusted person. These different reputations can be also represented by influencing the dumping factor. The higher the trustworthiness of someone in my interest group, who sends me her own context and rankings, the higher the probability to reach the resources in that set. Still, of course, even if two users exchange all of their context metadata, they still will not have the same rankings, as local usage information such as number of accesses etc., which influences rankings, always stays local and is not exchanged. Note that in our data graph, group members usually appear as instances of authors or as senders of emails [2], so we can use their rank as one possible indicator of their trustworthiness.

5   Related Work

[3] presents a class of model-based recommendation algorithms for creating a top-N list of recommendations. In their approach, they first determine the similarities between the various items and then use them to identify the set of items to be recommended. [3] also addresses the key steps of this class of algorithms: which are the methods used to compute the similarity between items and which are the methods used to combine

these similarities, in order to compute the similarity between a basket of items and a candidate recommender item. Opposed to this, in our approach the recommended items are based on user preferences.

Tapestry [5] is a recommender system which, in a sense, is similar to our approach. Tapestry is an e-mail filtering system, designed to filter e-mails received from mailing lists and newsgroup postings. Each user can write a comment / annotation about each email message and share these annotations with a group of users. A user can then filter these email messages by writing queries on these annotations. Though Tapestry allows individual users to benefit from annotations made by other users, the system requires an individual user to write complicated queries. We extend the idea in Tapestry by annotating not only emails but other resources on user's desktop. In addition, exchange of annotations is handled (semi-) automatically.

The first system that generated automated recommendations was the GroupLens system, [9]. The system, like in our case, provides users with personalized recommendations by identifying a neighbourhood of similar users and recommending the articles that this group of users finds interesting.

The most interesting work for recommendation infrastructures, which does not require a central recommender server is PocketLens, [7]. The paper discusses on how to preserve privacy in such an infrastructure. In contrast to our work, they do not exploit semantic connections between items, such as we have for citation relationships.

The Ringo [10] social networked system recommends music after the user's interests and preferences via e-mail. It is based on the idea that we have to rely on friends and other people whose judgement we trust to make recommendations to us. It makes use of social information filtering algorithms which help the system in making suggestions after the items that other users rated highly (people with similar tastes). It is designed as an on-line service to which users send e-mails with their ratings/preferences and in short time receive suggestions about what to listen in the future. This system builds a user profile as a vector of ratings and uses them with some formulas to compute the resemblance or disjunction of tastes. In our case we have groups that are defined around the same interest. It also uses a system of exchanging ratings between users and the centralized system and receiving related music.

Compared to the usual recommender systems, including the commercial ones such as Amazon.com, which usually suggest single items, we have the potential to make semantically richt suggestions that are represented as parts of a semantic network which we exchange. Additionally we also provide to the user information about other users' rankings. While most recommender systems define groups by relying on the overlap among preferred items, we rely on an explicit group membership denotation based on FOAF metadata.

6 Conclusions & Future Work

FOAF is a nice vocabulary to describe social networks, but most of the current applications are centered around describing social networks and not how to use them. This paper explores first steps on how to build upon FOAF and rich semantic web metadata to exchange and recommend context information and resources in a social network.

These contextual metadata are described by appropriate annotation ontologies, and are exchanged within FOAF groups as specified by the group members. The exchange of metadata is done by means of additional attachments for each document exchanged via email, extending email exchange from pure document exchange to an exchange of both document and relevant context information.

There are quite a few interesting issues to be investigated in future work, including privacy and security issues as well as the integration of our ranking algorithm with reputation measures. It is worthy to note that the ranking we compute for different resources can be compared to the ratings which are used in recommender systems. We can therefore not only share resources which are semantically connected to the ones we are exchanging, but also resources which are ranked/ rated highly by peers in our community. An additional interesting aspect is to explore dynamic social networks, where groups are not statically defined from the beginning but dynamically based on the exchange of context metadata. In this case users can initially choose which pieces of metadata information they want to append to a document for certain recipients, and common exchange patterns then determine common interest groups and allow automatic exchange of metadata based on these previous interactions.

References

1. A. Balmin, V. Hristidis, and Y. Papakonstantinou. Objectrank: Authority-based keyword search in databases. In *VLDB*, Toronto, September 2004.
2. P. Chirita, R. Gavriloaie, S. Ghita, W. Nejdl, and R. Paiu. Activity based metadata for semantic desktop search. In *Proceedings of the 2nd European Semantic Web Conference*, Crete, May 2005.
3. M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. In *ACM Transactions on Information Systems*, January 2004.
4. R. Gavriloaie, W. Nejdl, D. Olmedilla, K. Seamons, and M. Winslett. No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web. In *Proceedings of the 1st First European Semantic Web Symposium*, Heraklion, Greece, May 2004.
5. D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collabarative filtering to weave an information tapestry. In *ACM Press*, December 1992.
6. Teevan J., Alvarado C., Ackerman M. S., and Karger D. R. The perfect search engine is not enough: A study of orienteering behavior in directed search. In *CHI*, Vienna, April 2004.
7. B. N. Miller, J. A. Konstan, and J. Riedl. Pocketlens: Toward a personal recommender system. *ACM Trans. Inf. Syst.*, 22(3):437–476, 2004.
8. W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, and T. Risch. EDUTELLA: a P2P Networking Infrastructure based on RDF. In *In Proc. of 11th World Wide Web Conference*, Hawaii, USA, May 2002.
9. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM Press, 1994.
10. U. Shardanand and P. Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pages 210–217, 1995.
11. The foaf project. http://www.foaf-project.org/.