Personalized Information Access in a Bibliographic Peer-to-Peer System

Peter Haase[1] and **Marc Ehrig**[1] and **Andreas Hotho**[1] and **Björn Schnizler**[2]

[1]Institute AIFB
University of Karlsruhe, Germany
{pha, meh, aho}@aifb.uni-karlsruhe.de

[2]Information Management and Systems
University of Karlsruhe, Germany
schnizler@iw.uka.de

Abstract

The Bibster system is an application of the use of semantics in Peer-to-Peer systems, which is aimed at researchers that share bibliographic metadata. In this paper we describe the design and implementation of recommender functionality in the Bibster system which allows personalized access to the bibliographic metadata available in the Peer-to-Peer network. These functions are based on a semantic user profile which is created from content and usage information as well as a similarity function. Furthermore, these functions make use of the semantic topology of the Peer-to-Peer system.

Introduction

In recent years, the advantages of Peer-to-Peer architectures over centralized approaches have been advertised (Oram 2001), and to some extent realized in existing applications: no centralized server, robustness against failure of any single component, autonomy of the nodes, scalability both in data-volumes and number of connected parties. However, because of the lack of central control, the complexity of the system and the heterogeneity of the data, the use of semantics is crucial in this setting (Broekstra *et al.* 2003). For example, semantic descriptions of metadata can be used to cluster peers with similar content or interests to build semantic topologies (Nejdl *et al.* 2002), (Castano *et al.* 2003). These semantic topologies may reflect communities of interest or social networks and can then be exploited for example for efficient query routing as well as for personalization and adaptation.

The Bibster system[1] is such an application of the use of semantics in Peer-to-Peer systems (Broekstra *et al.* 2004). Bibster is aimed at researchers that share bibliographic metadata. Currently, many researchers in computer science keep lists of bibliographic metadata in BibTeX format, that they must laboriously maintain manually, for which they do not have an easy overview, and that has greatly varying quality. Many researchers own hundreds of kilobytes of bibliographic information, in dozens of BibTeX files. At the same time, many researchers are willing to share these resources, provided they do not have to invest work in doing so. We therefore also assume that the researchers do not provide malicious metadata, such that trust and security issues are not a problem[2].

Bibster enables the management of bibliographic metadata in a Peer-to-Peer fashion: it allows to import bibliographic metadata, e.g. from BibTeX files, into a local knowledge repository, to share and search the knowledge in the Peer-to-Peer system, as well as to edit and export the bibliographic metadata.

In this paper we describe the design and implementation of recommender functionality in the Bibster system which allows personalized access to the bibliographic metadata available in the Peer-to-Peer network according to the particular needs of the users.

These recommender functions build upon two main features of the Bibster system:

- *Semantic representation of metadata*: When bibliographic entries are made available for use in Bibster, they are structured and classified according to two bibliographic ontologies, the SWRC[3] ontology and the ACM[4] topic hierarchy. This ontological structure is then exploited to help the user formulate semantic queries. Query results again are represented according to the ontology. These semantic representations of the knowledge available on the peers, the user queries and relevant results allow us to directly create a semantic user profile and rich semantic similarity functions as a basis for recommending information that may potentially be interesting to the user.
- *Peer-to-Peer infrastructure with a semantic topology*: The Peer-to-Peer infrastructure reflects the distributed, decentralized and dynamic nature of creation of bibliographic metadata in a research community. In fact, a centralized solution does not exist and cannot exist, because of the multitude of informal workshops that researchers refer to, but that do not show up in centralized resources such as

[1]http://bibster.semanticweb.org/

[2]For other scenarios, where this assumption does not hold, we have developed a metadata model which covers trust, confidence, etc. (Broekstra *et al.* 2003)

[3]http://www.semanticweb.org/ontologies/swrc-onto-2001-12-11.daml

[4]http://www.acm.org/class/1988/

DBLP[5]. The decentralized Peer-to-Peer architecture can immediately be exploited for recommending newly created data as soon as it becomes available in the network.

Using semantic descriptions of the knowledge available on the peers, we are able to create semantic topologies that reflect the social networks of research communities: Peers with similar interests and expertise are clustered, such that the semantic neighborhood of a peer automatically covers a set of peers that contain relevant information for the specific community of interest. Furthermore, to address the cold start problem that recommender systems typically have to face (Schein *et al.* 2002), we can make use of the of the peer's semantic neighbors to create an initial user profile.

Example Scenarios

We will now illustrate the advantages of Bibster as a semantics-based, Peer-to-Peer recommender system with three typical usage scenarios, which we will use as a running example throughout the paper.

In the first scenario, suppose a researcher who is an expert on the topic of "Intelligent Agents" is searching for bibliographic metadata of new books about the topic "Artificial Intelligence"using the regular search functionality of the Bibster system. The corresponding query is routed in the semantic topology of the Peer-to-Peer network to the peers that may potentially return relevant answers. Among the results there may be an entry of the book "Handbook on Ontologies". Suppose the researcher considers this book relevant and saves the corresponding metadata into his local knowledge base. Now, the researcher might be interested in *similar publications*, which address similar topics or were written by a similar author constellation. Therefore the researcher could use the recommender function of Bibster to find similar entries – according to his definition of similarity – in the semantic neighborhood of the peer, again exploiting the semantic topology. The system might find the article "Knowledge Processes and Ontologies", which is about a subtopic of "Artificial Intelligence" and was written by a similar author constellation.

In the second scenario, the researcher might also want the system to proactively *recommend relevant publications* when they are available in the network. He could thus avoid searching the network manually in regular intervals. For example, the peer may have a semantic link to a special conference peer which provides the bibliographic metadata of conferences covering a certain set of topics, say a dedicated AAAI peer for the topic of "Artificial Intelligence". Without performing explicit queries, the researcher would be proactively provided with information about the new publications of his interest which were published at the relevant conferences. The recommender function can here exploit the area of expertise of the researcher, the queries he performed recently and the results that he considered relevant.

In a final scenario, the researcher may want to explore the semantic topology, e.g. to find *similar peers*. On the one hand, this information would make it possible to query this specific peer, e.g. a query for all journal items shared by this peer. On the other hand, the researcher could establish a personal contact to researchers interested in similar topics.

In the remainder of this paper we will first describe the design of the Bibster system. We will then present a model of ontology-based similarity for the bibliographic domain and the user profile, which are the basis for the recommender functions presented in the subsequent section. We will conclude after a discussion of related work.

Bibster - A Bibliographic Peer-to-Peer System

The Bibster system as described in (Broekstra *et al.* 2004) has been implemented as an instance of the SWAP System architecture as introduced in (Broekstra *et al.* 2003).

The Peer-to-Peer network consists of a set of peers P. In our bibliographic scenario, a peer $p \in P$ represents a researcher. Figure 1 shows a high-level design of the architecture of a single node in the Peer-to-Peer system.

We will now briefly present the individual components as instantiated for the Bibster system.

Knowledge Sources The knowledge sources in the Bibster system are sources of bibliographic metadata, such as BibTeX files stored locally in the file system of the user.

Knowledge Source Integrator The Knowledge Source Integrator is responsible for the extraction and integration of internal and external knowledge sources into the Local Node Repository. This task comprises (1) means to access local knowledge sources and extract a semantic representation of the available knowledge and (2) to integrate knowledge from remote peers. For the semantic extraction of bibliographic metadata from BibTeX files, we employ the BibToOnto[6] component. Knowledge of local and remote sources is merged using a semantic similarity measure to detect duplicate query results.

Local Node Repository In order to manage its information models and views as well as information acquired from the network, each peer maintains an internal working model stored in an RDF knowledge base, the Local Node Repository. This model provides the following functionality:

- Mediate between views and stored information
- Support query formulation and processing
- Specify the peer's interface to the network
- Provide the basis for peer ranking and selection

In the Bibster system, the Local Node Repository is based on the RDF-S Repository Sesame (J. Broekstra 2001). The query language SeRQL is used to formulate semantic queries against the Local Node Repository.

[5] http://dblp.uni-trier.de/

[6] http://bibtoonto.sourceforge.net/

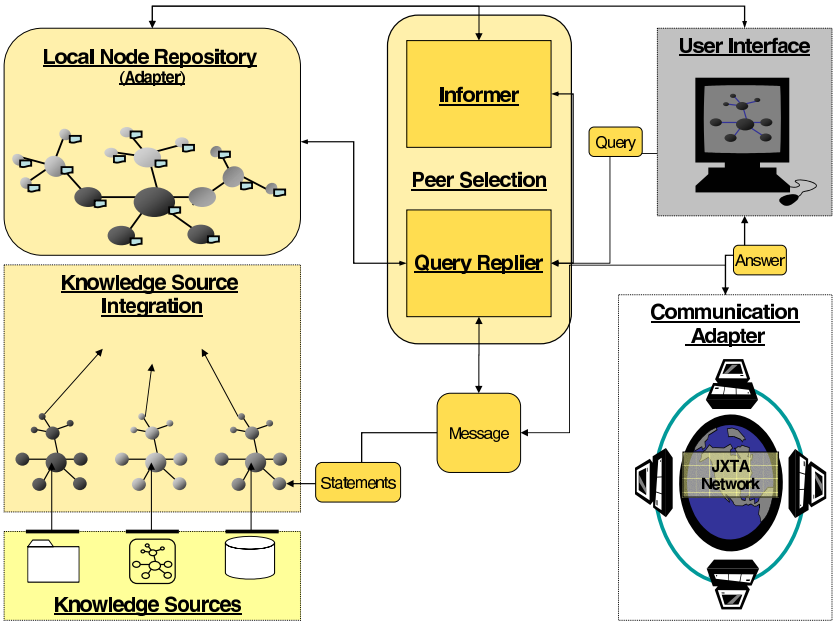

Figure 1: SWAP System Architecture

Informer The task of the Informer is to proactively advertise the available knowledge of a peer in the Peer-to-Peer network and to discover peers with relevant knowledge that may be relevant for answering the user's queries. This is realized by sending advertisements about the expertise of a peer. In the Bibster system, these expertise descriptions contain a set of topics that the peer is an expert in. Peers may accept – i.e. remember – these advertisements, thus creating a semantic link to the other peer. These semantic links form a semantic topology, which is the basis for intelligent query routing.

Query Replier The Query Replier is the coordinating component which controls the process of distributing queries. It receives queries from the user interface and distributes them according to the content of the query. When the peer receives a query from another peer, it tries to answer or forward it. Based on the knowledge about the expertise of other peers, it is decided to which peers a query should be sent.

User Interface The user interface, as shown in figure 2 allows the user import, create and edit bibliographic metadata as well as to formulate queries in an intuitive manner. In addition to simple keyword based queries against all attributes, the user can formulate advanced semantic queries against the SWRC ontology and the ACM topic hierarchy.

Furthermore, the scope of the query can be specified: Queries can be evaluated on the local peer, on selected peers, or globally. The query results, which are visualized in a list grouped by duplicates, can then be integrated into the local repository or exported in formats such as BibTeX and HTML.

Communication Adapter This component is responsible for the network communication between peers. It serves as a transport layer for other parts of the system, for sending and forwarding queries. It hides and encapsulates all low-level communication details from the rest of the system. In the specific implementation of the Bibster system we use JXTA as the communication platform.

Semantic Topologies and Query Routing in Bibster

As mentioned above, the knowledge of the peers about the expertise of other peers forms a semantic topology which is the basis for intelligent query routing.

The model and evaluation of expertise based peer selection using semantic topologies has been described in detail in (Haase, Siebes, & van Harmelen 2004).

The expertise of a peer is an abstract description of the knowledge available in the local repository. In the bibliographic scenario, the expertise is a set of topics. Peers promote their expertise in the network by sending advertisements, which effectively associate a peer with an expertise. The semantic topology can then be described by the following relation:

$Knows \subseteq P \times P$, where $(p_1, p_2) \in Knows$ means that peer p_1 knows about the expertise of peer p_2.

The relation $Knows$ is established by the selection of which peers a peer sends its advertisements to. Furthermore peers can decide to accept an advertisement, e.g. to include it in their registries, or to discard the advertisement.

The peer selection algorithm extracts *subjects* from *user queries* and matches these subjects against the known expertise descriptions using a similarity functions. The queries are then routed to the peers whose expertise best matches the subject of the query.

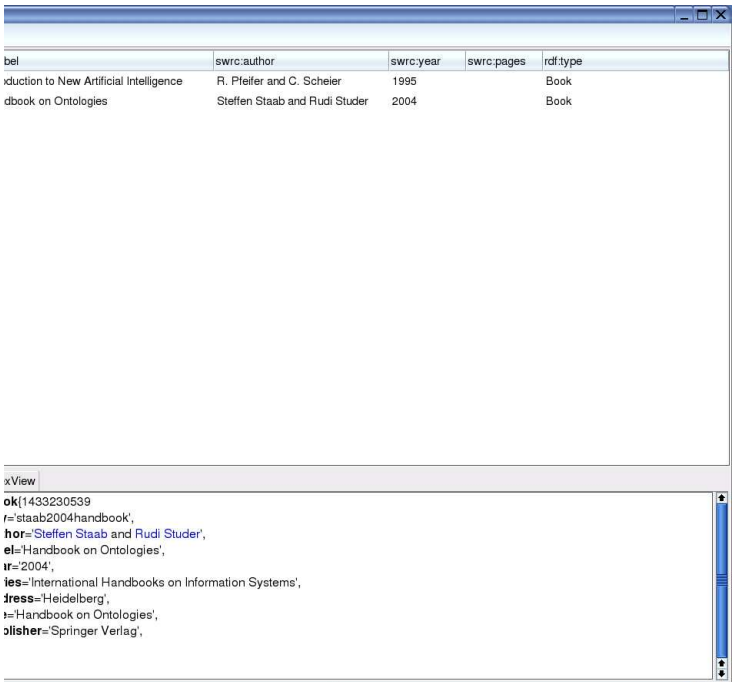

Figure 2: User interface for the Bibster application

The properties of the topology and the peer selection algorithm considerably affect the performance of query routing. Best results can be achieved, if the peers are clustered according their expertise. In (Haase, Siebes, & van Harmelen 2004) we have shown how an effective semantic topology can be created by remembering advertisements from peers that have a semantically similar expertise. More advanced algorithms cluster peers in a decentralized manner by "rewiring": Here some semantic links are dynamically replaced with links to more similar peers, which are found using *random walk* or *gradient walk* strategies.

Ontology Based Similarity

In this section we will first describe the bibliographic ontologies employed in the Bibster system. Subsequently, we will define a semantic similarity function for this bibliographic ontology, which serves as the basis for the recommender functions presented in a following section.

The Bibliographic Ontologies

In our bibliographic scenario we make use of two common ontologies:

The first ontology is the Semantic Web Research Community Ontology (SWRC), which models among others a research community, its researchers, topics, publications, and properties between them (Handschuh, Staab, & Maedche 2001). The SWRC ontology defines a shared and common domain theory which helps users and machines to communicate concisely and supports exchange of semantics. The

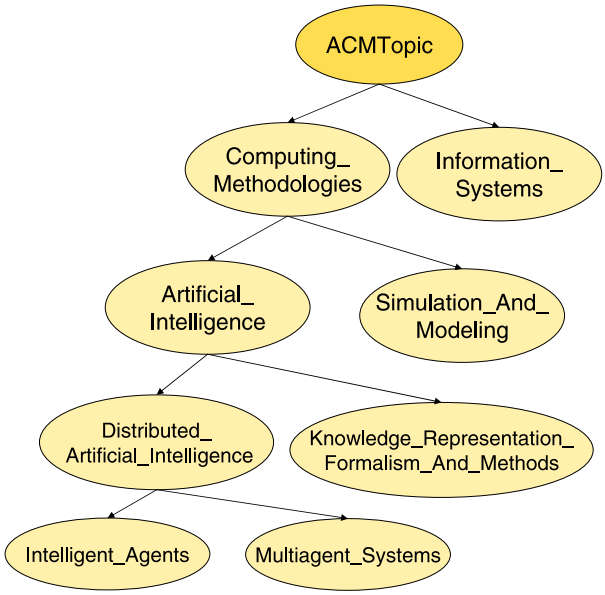

Figure 3: Fragment of the ACM Topic Hierarchy

second ontology is the ACM topic hierarchy. It describes specific categories of literature for the Computer Science domain, covering 1287 topics. Figure 3 shows a small fragment of the hierarchy relevant for our example scenarios. In addition to the sub- and super-topic relations, the hier-

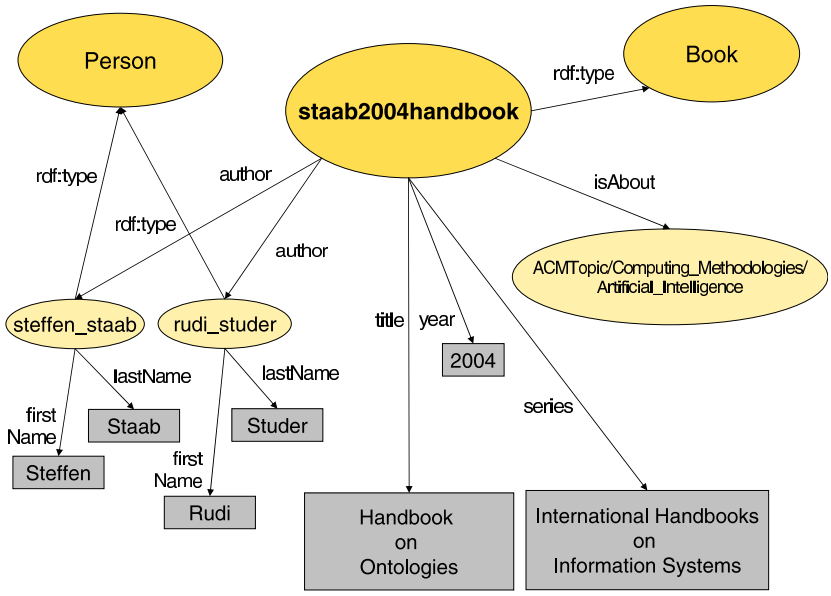

Figure 4: RDF graph for Example 1

archy also provides information about related topics. The topic hierarchy therefore provides a quick content reference and assists users in searching for related publications. In the context of a recommender system this classification is crucial for identifying similarities.

Bibliographic entries that a user made available to Bibster are described using these two ontologies. The classification according to the ACM ontology is initially done automatically using lexical matching of the topic labels against the titles of the publications. Additionally, it is possible to reclassify the entries manually in the user interface of Bibster.

The ontologies and the specific bibliographic instance data are represented in RDF.

The following example shows a fragment of a sample bibliographic item based on the SWRC ontology. Figure 4 illustrates the example as an RDF graph.

Example 1

```
<swrc:Person rdf:about="urn://urn.jxta.uuid#371003986">
 <rdfs:label>Steffen Staab</rdfs:label>
 <swrc:name>Steffen Staab</swrc:name>
 <swrc:firstName>Steffen</swrc:firstName>
 <swrc:lastName>Staab</swrc:lastName>
</swrc:Person>

<swrc:Person rdf:about="urn://urn.jxta.uuid#233551477">
 <rdfs:label>Rudi Studer</rdfs:label>
 <swrc:name>Rudi Studer</swrc:name>
 <swrc:firstName>Rudi</swrc:firstName>
 <swrc:lastName>Studer</swrc:lastName>
</swrc:Person>

<swrc:Book rdf:about="urn://urn.jxta.uuid#1433230539">
 <rdfs:label>Handbook on Ontologies</rdfs:label>
 <swrc:key>staab2004handbook</swrc:key>
 <swrc:title>Handbook on Ontologies</swrc:title>
 <swrc:author rdf:resource="urn://urn.jxta.uuid#371003986"/>
 <swrc:author rdf:resource="urn://urn.jxta.uuid#233551477"/>
 <swrc:year>2004</swrc:year>
 <swrc:address>Heidelberg</swrc:address>
 <swrc:publisher>Springer Verlag</swrc:publisher>
 <swrc:series>
  International Handbooks on Information Systems
 </swrc:series>
 <swrc:isAbout rdf:resource="http://daml.umbc.edu/
   ontologies/topic-ont#ACMTopic/
   Computing_Methodologies/Artificial_Intelligence"/>
</swrc:Book>
```

Authors and editors are represented as instances of the `swrc:Person` class. They can be identified by their unique URNs, which are constructed in a special way to comply with the requirements of the JXTA network infrastructure.

The publication itself is instantiated as a `swrc:Book`, which is a subclass of `swrc:Publication`. The ACM topics corresponding to the publications are represented with the `swrc:isAbout` properties. In this example the associated topic is `Artificial Intelligence`.

Semantic Similarity

We will now first describe our notion of similarity we use in our recommender system. Then we will present individual similarity functions and show how to combine these. Some of the measures are generic similarity functions independent of a specific domain ontology. However, using background knowledge about the bibliography domain allows to define more specific similarity functions.

Similarity Function A similarity function for RDF resources R of a knowledge base is a function

$$sim : R \times R \rightarrow [0..1]$$

with the properties as presented in (Bisson 1995). This function is based on different features of the respective resources. Individual functions for each feature are combined using an aggregation function to compute an overall similarity result.

Features Each resource type is compared based on specific features. For persons and organizations we rely solely on the their names, whereas for publications we use a wide range of features: title, publication type, authors and editors, publisher, institute and university, booktitle or journal with the series number and address, page numbers, publication year, and the ACM topic the publication was classified to.

Individual Similarity Functions For these individual features we use specific functions, which do not only determine the similarity on the syntactic level, but also consider the semantics of the ontological structures. The individual functions take the following characteristics of the ontology into account:

- *Data Value Layer*, where we consider the atomic data values of the attributes of the instances, which in RDF are represented as typed literals,
- *Graph Layer*, where we consider relations between the RDF resources,
- *Ontology Layer*, where we consider ontological information, such as the class hierarchy,
- *Domain Specific Knowledge*, where we use domain specific features with corresponding heuristics.

We will now present the specific methods used for the bibliographic ontology:

Data Value Layer: To determine the similarity of data values d_1, d_2 of type string (e.g. to compare the names of persons) we use the syntactic similarity sim_{syn} of (Maedche 2001). It relies on the edit distance (ed) of (Levenshtein 1966), which basically determines how many atomic actions as character addition or deletion are required to transform one string into the other one.

$$sim_{syn}(d_1, d_2) = max(0, \frac{min(|d_1|, |d_2|) - ed(d_1, d_2)}{min(|d_1|, |d_2|)})$$

Graph Structure Layer A publication resource is structurally linked with person resources, e.g. authors. Thus we can compare two publications on the basis of the similarity of the sets of authors. To compare the similarity of two sets of resources E and F, we average over the similarities of the resources of the one set with the most similar resource of the respective other set:

$$sim_{set}(E, F) = \frac{\sum_{e \in E} \max_{f \in F} sim(e, f) + \sum_{f \in F} \max_{e \in E} sim(f, e)}{|E| + |F|}$$

Ontology Layer One possible generic function to determine the semantic similarity of concepts in a concept hierarchy (such as topics in the ACM topic hierarchy) has been presented by (Rada *et al.* 1989):

$$sim_{taxonomic}(c_1, c_2) = \begin{cases} e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}, & \text{if } c_1 \neq c_2, \\ 1, & \text{otherwise} \end{cases}$$

$\alpha \geq 0$ and $\beta \geq 0$ are parameters scaling the contribution of shortest path length l and depth h in the concept hierarchy, respectively. The shortest path length is a metric for measuring the conceptual distance of c_1 and c_2. The intuition behind using the depth of the direct common subsumer in the calculation is that concepts at upper layers of the concept hierarchy are more general and are semantically less similar than concepts at lower layers. Complying with (Rada *et al.* 1989), for the comparison of ACM topics the parameters are set to $\alpha = 0.2$, $\beta = 0.6$.

Domain Specific Knowledge In the SWRC domain ontology there are many subconcepts of publications: articles, books, and technical reports to just name a few. We know that if the type of a publication is not known, it is often provided as Misc (e.g. in Citeseer[7]).
Instead of using a generic similarity function, we can thus define:

$$sim_{type}(c_1, c_2) = \begin{cases} 1, & \text{if } c_1 = c_2, \\ 0.75, & \text{if } (c_1 = \text{Misc} \vee c_2 = \text{Misc}) \\ 0, & \text{otherwise} \end{cases}$$

Experiments with sample data have shown that a similarity value of 0.75 yields meaningful results if one of the publications is of type Misc.

Another domain specific function is used for the similarity between years. The closer the years of the publications are, the higher their similarity:

$$sim_{year}(year_1, year_2) = \frac{1}{1 + |year_1 - year_2|}$$

Aggregated Similarity Function Based on the individual similarity functions, an overall value can be obtained for example using a weighted average function

$$Sim_W(i_1, i_2) = \frac{1}{\sum_{k=1}^{n} w_k} \sum_{k=1}^{n} w_k sim_k(i_1, i_2)$$

with w_k being the weight for a specific function sim_k. Because of the semi-structured nature of bibliographic metadata, some attributes may not be provided such that some individual measures may not apply. Therefore, for non-mandatory attributes, the weight w_k will be adjusted to 0 if either one of the compared resources does not provide the attribute.

Example 2 We now present a complete example of a combined similarity function for the bibliographic scenario, in which we compute the semantic similarity of the publication p_1 from example 1 with the publication p_2 as shown in figure 5. When comparing the two example publications

[7] http://citeseer.nj.nec.com/

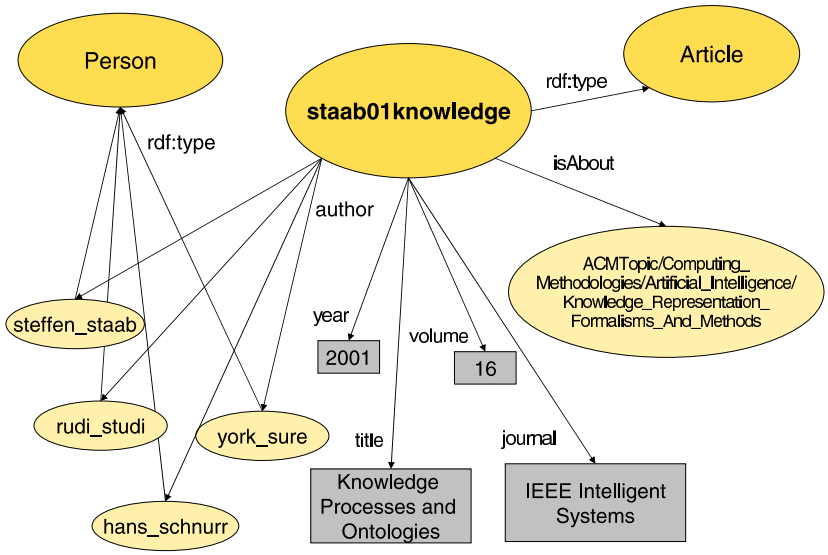

Figure 5: RDF graph for Example 2

applying the similarity functions from above we obtain:

$sim_1(p_1, p_2) = sim_{type}(Book, Article) = 0$
$sim_2(p_1, p_2) = sim_{syn}$(*"Handbook on Ontologies", "Knowledge Processes and Ontologies"*) $= 0.14$
$sim_3(p_1, p_2) = sim_{taxonomic}(Artificial\_Intelligence, Knowledge\_Representation\_Formalisms\_And\_Methods) = 0.98$
$sim_4(p_1, p_2) = sim_{set}$(*(steffen_staab,rudi_studer), (steffen_staab,rudi_studer,hans_schnurr,york_sure)*) $= 0.67$
$sim_5(p_1, p_2) = sim_{year}(2004, 2001) = 0.25$

In our example, we use a weight-vector of $W = (2, 2, 9, 9, 2)$*, which prefers the topic and the author attributes over the rest of the attributes:*

$$Sim_W(p_1, p_2) = \frac{1}{\sum_{k=1}^{n} w_k} \sum_{k=1}^{n} w_k sim_k(p_1, p_2) = 0.65$$

The similarity value indicates the similarity of the resources and can be used directly as a rank value. With an assumed threshold for similarity of 0.5, the publication p_2 would be considered similar to p_1.

Semantic User Profile

The user profile is built on the basis of the semantic representation of the shared knowledge (content) and usage information. Conforming with the model presented in (Montaner, Lopez, & De La Rosa 2003), we will now describe the user profile representation, the initial user profile and profile adaptation.

User Profile Representation

Definition 1 *A* user profile *is a structure* $PR := (E, Q, R, W, t)$ *consisting of*

- *the expertise description* E*,*
- *a set of recent queries* Q*,*
- *a set of recent relevant instances* R*,*
- *a structure* W *which defines the weights for the similarity function,*
- *a threshold* $t \in [0, 1]$*.*

We will now describe the roles of the elements of a user profile.

Expertise E: The expertise E is a set of topics which the user is knowledgeable about. It is built on the assumption that if a user has a knowledge base with bibliographic items about certain topics, he is a researcher with a certain expertise and interests in these topics. Consequently, he might be interested in other bibliographic items about these topics that are not available in his local knowledge base. The expertise model is constructed directly from the knowledge base of the peer: It comprises all topics for which the knowledge base contains classified instances. In this sense, the expertise model can be seen as an abstraction or index structure of the knowledge base.

The expertise model can easily be extended to not only cover topics, but also for example certain conferences, authors, etc.

Recent queries Q: The queries are an important part of the interaction of the user with the system that reflect the information need and interest of the user. To exploit this knowledge about the interest, we store a set of recent queries as part of the user profile. However, during the transformation of the information need into a query, information may get lost. In this sense, the user might be interested in instances that may not exactly match the query, but are semantically similar. Another reason to remember recent queries

is the following: It may be possible that at the time of querying no entries match a given query, either because matching entries do not exists or the relevant peer is currently offline. However, at a later point in time, matching entries could possibly be found.

As mentioned before, in Bibster we use the SeRQL query language. In our scenario, a researcher is looking a book about Artificial Intelligence. The user specifies his search request through the user interface as shown in the previous section. Internally, this request is formulated as a SeRQL query that looks as follows:

Example 3

```
construct distinct
    {s} prop {val}
from
    {s} <rdf:type> {<swrc:Book>};
        <swrc:isAbout> {<acm:ACMTopic/Computing_Methodologies/
                         Artificial_Intelligence>}
```

Instead of storing the SeRQL query itself, for each query $q \in Q$ we store the corresponding attribute value pairs that the user specified as an RDF resource:

Example 4

```
<rdf:Description rdf:about="query1">
 <rdf:type rdf:resource=
  "http://www.semanticweb.org/ontologies/
   swrc-onto-2001-12-11.daml#Book"/>
 <swrc:isAbout rdf:resource=
  "http://daml.umbc.edu/ontologies/topic-ont#ACMTopic/
   Computing_Methodologies/Artificial_Intelligence"/>
</rdf:Description>
```

This set of attribute value pairs is thus represented in the same way as the specification of a publication itself. We can therefore apply the semantic similarity measures defined before to calculate how close a bibliographic instance matches a query.

When considering the set of recent queries, we may be able to recommend items that may not have matched any of the queries exactly, but are semantically very close to the information need of user.

Recent Relevant Results R: The recent relevant results are a set of bibliographic instances obtained from previous searches. Here a bibliographic instance is considered relevant if it has been included into the local knowledge base by the user. We thus do not store all results that match the user query, instead we only store those instances that were subjectively relevant to the user. The recent relevant results therefore better reflect the information need of the user.

Weights W: Just as relevance, also similarity is a very subjective measure. We therefore allow to adjust the weights of the similarity function presented in the previous section as part of the user profile. By adjusting these weights, the user can specify which attributes of a the bibliographic metadata are more important for determining the relevance of similar items. For example, when the user requests the system to recommend publications similar to an existing one, he might indicate that he does not care about similarity of the title, but is interested in a similar constellation of co-authors. Another option might be that the user is interested in publications at similar conferences in the same or close year.

Threshold t: With the threshold t the user can specify how closely a resource must match the profile to considered relevant by the recommender functions. The threshold is used to filter the ranked results of the similarity functions. An increasing threshold will result in a more selective matching. The user can thus use the threshold to influence the result size: Depending on the amount of data available, the threshold may be increased or decreased to obtain a useful result size.

Initial User Profile and Profile Adaptation

We will now describe how the initial user profile is created and adapted. It is important to note how the cold start problem is addressed: We use a combination of content and usage based information in the user profile. In typical recommender systems, the cold start problem is caused by the initially unavailable usage information. In our system, we make use of the properties of the Peer-to-Peer network: Instead of starting with an empty profile, we reuse the profiles of similar peers in the semantic neighborhood.

Expertise: The initial expertise description E will be, as defined, the set of topics for which classified instances exist in the knowledge base KB of the peer. The expertise profile of the user is adapted whenever the knowledge base of the peer is updated. This means that whenever publications covering certain topics are added or removed from the knowledge base, the expertise description is updated accordingly.

Recent queries: To avoid to start with an empty set of recent queries, we start out with a sample of queries that were recently performed by the peers in the neighborhood (given by the semantic topology) of the peer.
When the user performs a query, it is added to the set of recent queries. As only n recent queries are stored, old queries are gradually forgotten. This could be done by always remembering the last n queries (FIFO), however, by temporary changes in the query behavior, the previous profile may get lost. Therefore, the items to be removed from the set of recent queries are selected randomly.

Recent Relevant Results: As for the recent queries, for the initial set of relevant results we also rely on the the profile of the peers in the semantic neighborhood and use a sample of results that were recently considered relevant by other peers. Whenever the user decides to store a bibliographic item to its local knowledge base, it is added to the list of recent relevant results. With the recent relevant results we realize an implicit relevance feedback: The recommended entries that were considered relevant by the user are immediately added to the user profile. For the gradual forgetting

of results, the same mechanisms are applied as described above for the recent queries.

Weights: As initial weights W for the similarity function, default values are used. For the Bibster system, useful default values have been determined heuristically with experiments. The weights of the similarity function can be adapted

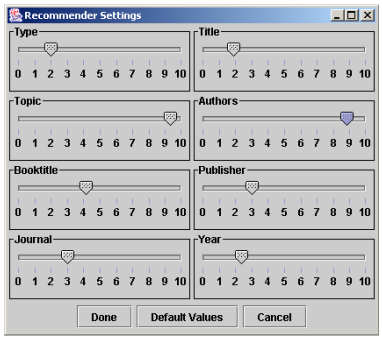

Figure 6: Slider Settings

manually by the user as shown in figure 6. Using the relevance feedback mechanisms described above, the weights could also be adjusted automatically by the system.

Threshold: The threshold is initially set to 0.5, which is a more or less arbitrary choice. The threshold can be increased or decreased by the user to affect the amount of recommended information. Alternatively, the threshold could be adjusted automatically by the system such that the amount of recommended data is kept manageable.

Recommender Functions

In this section we will explain how the three scenarios motivated in the introduction of this paper are realized in Bibster. The recommendation functions are realized using the similarity functions and user profiles presented before.

Recommending Similar Items

The first recommender function allows to provide the user with bibliographic entries that are semantically similar to relevant results from previous searches. In a typical browsing scenario the user may start out with a vague idea of his search criteria. The information need may not have been transformed into the correct query. Thus the available relevant content may not necessarily match the specified search. By recommending entries that are similar to the results that the user has considered as relevant, the quality of the search can be improved. As motivated before, similar may mean various things depending on the user context. Therefore, the weights of similarity function as part of the user profile are taken into account.

The set of relevant results, R, and the weights of the similarity function, W, are routed in the Peer-to-Peer network as a special request to the potentially relevant peers. The remote peers evaluate the request against their local knowledge base KB using the recommendation function defined as follows:

$$Rec_1(KB, R, W, t) := \{i \in KB | \exists r \in R : Sim_W(i, r) \geq t\}$$

The function thus returns the set of bibliographic items from the instance set of the knowledge base whose semantic similarity with one of the relevant results is greater than the defined threshold t, determined by the specified weights W. The set of similar entries is then sent back and presented to the user.

Example 5 In our scenario, the user has performed the query for books on the topic of "Artifical Intelligence", as shown in example 3. He may have selected the "Handbook on Ontologies" from example 1 as a relevant result. The peer selection algorithm may have routed the request for similar publications to a remote peer because of its expertise in "Knowledge Representation Formalisms And Methods". This peer's knowledge base KB contains the entry "Knowledge Processes and Ontologies" from figure 5, which would be recommended to the originating peer based on the calculation shown in example 2.

Recommending Potentially Relevant Items

Unlike the previous function, which required a set of relevant results to be identified by the user, this function proactively recommends potentially relevant items based solely on the user profile.

Analogously to the previous function, the Profile, $PR := (E, Q, R, W, t)$, is propagated as a special request in the Peer-to-Peer network and the function Rec_2 is evaluated against the knowledge base KB of the remote peer:

$$\begin{aligned} Rec_2(KB, PR) = \{i \in KB | \exists q \in Q, \exists r \in R : \\ \mathcal{A}(Sim_{Topics}(E, Topics(i)), \\ Sim_W(i, q), \\ Sim_W(i, r)) \geq t\} \end{aligned}$$

The recommender function Rec_2 computes a combined relevance of bibliographic items based on the individual elements of the user profile, i.e. expertise, recent queries and recent relevant results. It uses an aggregation function $\mathcal{A}$ that determines the overall similarity as a composition of the individual similarity measures. In the easiest case, the aggregation function $\mathcal{A}$ is again simply a weighted average.

The following individual similarity measures are used: For the expertise we determine the similarity of the set of expertise topics with the set of topics for which the instances i of the knowledge base are classified for (determined by the $Topic(i)$ function) using the function Sim_{Topics} (based on $sim_{taxonomic}$ presented above). We then compute the weighted similarity of the recent queries we with the instances of the knowledge base. Here the queries are treated as instances, as described above. Similarly, the similarity of the recent relevant results with the instances of the knowledge base is determined.

The relevant results are returned to the querying peer.

Example 6 Continuing with our scenario, the user profile now consists of the following: The user's expertise is the topic of "Intelligent Agents", the user recently performed the query from example 3 and considered the result "Handbook on Ontologies" as relevant. Suppose the user's peer is connected in the semantic topology to a peer that covers publications published at AAAI conferences and associated workshops. Among these publications is for example "Towards Evaluation of Peer-to-Peer-based Distributed Knowledge Management Systems" (Ehrig et al. *2003), which was co-authored by one of the authors of the recently relevant "Handbook on Ontologies" and is classified to be about "Multiagent Systems", a topic similar to "Intelligent Agents". This publication would therefore match the user profile and would be recommended as relevant. For space constraints we omit the complete calculations of the recommender function.*

Recommending Similar Peers

This last recommender function allows to find peers in the network with a similar expertise E. Unlike the two previous functions, Rec_3 can be evaluated on the local peer, as the advertisements of the peers' P expertise are already known to the local peer:

$$Rec_3(p_1) := \{p_2 \in P | (p_1, p_2) \in Knows\}$$

The function is implicitly realized by the semantic topology of the Peer-to-Peer network: The similar peers are all those peers that have a link in the semantic topology, which was created because their expertise is semantically similar, determined by a threshold t:

$$Knows(p_1, p_2) \implies$$
$$Sim_{Topics}(Expertise(p_1), Expertise(p_2)) \geq t$$

(Here $Expertise(p)$ returns the set of topics that the peer has advertised.)

The semantic topology thus is not only used for efficient peer selection and query routing, but also enables the user to find similar peers in the Peer-to-Peer network. The user can then, for example, address queries directly to relevant peers.

Example 7 The user from our scenario may now be interested in exploring the semantic topology to find peers with a similar expertise. As stated in the previous example, among these peers may be the dedicated AAAI peer, because of the similarity of the topics covered by AAAI and the expertise of the user. Knowing of the existence of this special peer, the user could now direct specific queries directly to the peer, for example to retrieve the complete proceedings of the workshop from the previous example.

Related Work

Several research areas are relevant for our discussion of related work: We will first discuss related semantics-based Peer-to-Peer systems. Secondly, we consider the research done in the field of ontology based recommender systems and their application to personalized information access. Furthermore, our approach to compute the similarity measures and also our recommender approach is very similar to principle approaches known from cased based reasoning systems (CBR) (cf. (Hayes, Cunningham, & Smyth 2001)).

The use of semantics in Peer-to-Peer systems focuses mainly on two problems: Improving the efficiency of query routing and the support for local, individual schemas instead of global schemas. Edutella (Nejdl *et al.* 2002) is a Peer-to-Peer system based on the JXTA platform, which offers very similar base functionality as the SWAP system. (Nejdl & others 2003) present schema-based Peer-to-Peer networks and the use of super-peer based topologies for these networks, in which peers are organized in hypercubes. (Löser & others 2003) show how this schema-based approach can be used to create Semantic Overlay Clusters in a scientific Peer-to-Peer network with a small set of metadata attributes that describe the documents in the network. In contrast, the approach in our system, is completely decentralized in the sense that it does not rely on super-peers. (Ahlborn, Nejdl, & Siberski 2002) describe the design of a Peer-to-Peer network for open archives, where data providers, i.e. research institutes, form a Peer-to-Peer network which supports distributed search over all the connected metadata repositories. This scenario is similar to our bibliographic Peer-to-Peer scenario, however, their system has not been implemented up to this point.

(Tempich, Staab, & Wranik 2004) propose a new algorithm for semantic query routing – REMINDIN' – based on social metaphors. It would be interesting to see how these social metaphors could improve personalized information access.

Various systems address the issue of heterogeneity in Peer-to-Peer systems on the schema level, such as the Piazza peer data management system (Tatarinov *et al.* 2003), which allows for information sharing with different schemas relying on local mappings between schemas. While this can be seen as a form of personalization, none of the Peer-to-Peer systems address personalization in the sense of recommendations.

On the other hand, the research field of recommender systems is heavily active. A lot of different approaches and systems exist. Following the taxonomy of recommender systems in (Montaner, Lopez, & De La Rosa 2003) our system contains a semantics based profile, without profile learning techniques, using implicit relevance feedback. The profile adaptation takes place through adding new items and a gradual forgetting function. The most relevant part in the field of recommender systems are content based, especially ontology based, recommender systems, as our knowledge base represents the content of a peer.

(Middleton, Roure, & Shadbolt 2003) describe the improvement of classical recommender systems with ontologies. They use the ontology to enhance the user interface, to reduce the staring effort (Middleton *et al.* 2002) and to improve the recommendation accuracy (Middleton, Roure, & N.R.Shadbolt 2001). The approaches were tested on two user groups where the recommender system recommends research paper. Theses works shows the benefits of ontologies for recommender systems. Our approach is also based in

ontologies but we use a peer to peer and not a central server system.

(Amato & Straccia 1999) discuss the relevance of user profiles to model the information need of users and to personalize the access. Bibster captures the information for one peer in a similar way as it also derives a profile for every peer. (Schwarzkopf 2004) describes how the user interaction of a semantic portal can be improved by utilizing personal knowledge bases, which express semantic properties. This utilization is similar to the exploitation of the expertise obtained from the user's knowledge base in our user profile. (Dolog *et al.* 2003) present a rule-based approach to personalization in adaptive educational hypermedia systems, where the user's current knowledge state is used as the user profile and relevant content is determined using FOL rules. Bibster can be compared with adaptive hypermedia systems in the sense that relevant RDF-subgraphs are presented to the user using semantic similarity measures.

Conclusion

In this paper, we have described the design and implementation of recommender functionality in Bibster, a semantics-based Peer-to-Peer system for the exchange of bibliographic metadata between researchers.

We have presented a semantic similarity function for a bibliographic ontology, based on which we are able to match the bibliographic metadata against user profiles. These user profiles are built from content and usage information. We have shown how three specific recommendation functions are realized to recommend similar bibliographic entries, to proactively provide potentially relevant entries and to find similar peers in the Peer-to-Peer network. Further, we have shown how the semantic topology of the Peer-to-Peer network is used to route requests and efficiently find the relevant content as well as to address the cold start problem.

In order to measure the effectiveness of our approach, we are planning to execute an extensive evaluation study. As we cannot report on the results of this study at the time of writing, we present our evaluation plan.

Evaluation plan

The methods and functions of the Bibster system will be evaluated by means of a case study among the potential end users of the system. The participants of the study will use the Bibster system in their daily work. The study will begin with a core group of researchers representing a wide spectrum of research areas in Computer Science and which represent different levels of research experience. In a first step the number of participants will be approximately 50–60 persons. In the next step the system will be made available for the public on the project website.

On the one hand the case study will evaluate the user satisfaction to measure the particular benefit from the ontology based Peer-to-Peer based. Likewise we will evaluate the user satisfaction of the recommendations of our system. This evaluation will be done using user questionnaires.

On the other hand the technical aspects of the Bibster system will be evaluated through automated data collecting, i.e. recording and analyzing user and system activity by means of log files. For example, we log automatically the action when a user accepts a recommended bibliographic item. The log files are created locally on each peer and periodically sent to a central server. The gathered log files are then aggregated to allow overall evaluation.

Acknowledgments

Research reported in this paper has been partially financed by the EU in the IST projects SWAP (IST-2001-34103)[8] and SEKT (IST-2003-506826) (`http://www.sekt-project.com/`). We would like to thank our colleagues for fruitful discussions.

References

Ahlborn, B.; Nejdl, W.; and Siberski, W. 2002. OAI-P2P: A peer-to-peer network for open archives. In *Workshop on Distributed Computing Architectures for Digital Libraries - ICPP2002*.

Amato, G., and Straccia, U. 1999. User profile modeling and applications to digital libraries. In Abiteboul, S., and Vercoustre, A.-M., eds., *Proc. 3rd European Conf. Research and Advanced Technology for Digital Libraries, ECDL*, 184–197. Springer-Verlag.

Bisson, G. 1995. Why and how to define a similarity measure for object based representation systems. *Towards Very Large Knowledge Bases* 236–246.

Broekstra, J.; Ehrig, M.; Haase, P.; van Harmelen, F.; Kampman, A.; Sabou, M.; Siebes, R.; Staab, S.; Stuckenschmidt, H.; and Tempich, C. 2003. A metadata model for semantics-based peer-to-peer systems. In *Proceedings of the WWW'03 Workshop on Semantics in Peer-to-Peer and Grid Computing*.

Broekstra, J.; Ehrig, M.; Haase, P.; van Harmelen, F.; Menken, M.; Mika, P.; Schnizler, B.; and Siebes, R. 2004. Bibster - a semantics-based bibliographic peer-to-peer system. In *Proceedings of the WWW'04 Workshop on Semantics in Peer-to-Peer and Grid Computing*.

Castano, A.; Ferrara, S.; Montanelli, S.; Pagani, E.; and Rossi, G. 2003. Ontology-addressable contents in p2p networks. In *Proceedings of the WWW'03 Workshop on Semantics in Peer-to-Peer and Grid Computing*.

Dolog, P.; Henzen, N.; Nejdl, W.; and Sintek, M. 2003. Towards the adaptive semantic web. In *1st Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR'03)*.

Ehrig, M.; Schmitz, C.; Staab, S.; Tane, J.; and Tempich, C. 2003. Towards evaluation of peer-to-peer-based distributed knowledge management systems. In van Elst, L.; Dignum, V.; and Abecker, A., eds., *Proceedings of the AAAI Spring Symposium "Agent-Mediated Knowledge Management (AMKM-2003)"*, Springer LNAI. Stanford, California: Stanford University.

Haase, P.; Siebes, R.; and van Harmelen, F. 2004. Peer selection in peer-to-peer networks with semantic topologies.

[8] http://swap.semanticweb.org

In *International Conference on Semantics of a Networked World: Semantics for Grid Databases, 2004, Paris*.

Handschuh, S.; Staab, S.; and Maedche, A. 2001. Cream - creating relational metadata with a component-based. In *Proceedings of the First International Conference on Knowledge Capture K-CAP 2001*.

Hayes, C.; Cunningham, P.; and Smyth, B. 2001. A case-based reasoning view of automated collaborative filtering. In Aha, D. W., and Watson, I., eds., *Case-Based Reasoning Research and Development, 4th International Conference on Case-Based Reasoning, ICCBR 2001, Vancouver, BC, Canada, July 30 - August 2, 2001, Proceedings*, volume 2080 of *Lecture Notes in Computer Science*, 234–248. Springer.

J. Broekstra, A. Kampman, F. v. H. 2001. Sesame: An architecture for storing and querying rdf data and schema information. In D. Fensel, J. Hendler, H. L., and Wahlster, W., eds., *Semantics for the WWW*. MIT Press.

Levenshtein, I. V. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*.

Löser, A., et al. 2003. Efficient data store discovery in a scientific P2P network. In Ashish, N., and Goble, C., eds., *Proc. of the WS on Semantic Web Technologies for Searching and Retrieving Scientific Data*, CEUR WS 83.

Maedche, A. 2001. Comparing ontologies - similarity measures and a comparison study. Technical report, Forschungszentrum Informatik, Karlsruhe, Germany.

Middleton, S. E.; Alani, H.; Shadbolt, N.; and Roure, D. D. 2002. Exploiting synergy between ontologies and recommender systems. In Frank, M.; Noy, N.; and Staab, S., eds., *Proceedings of the WWW2002 International Workshop on the Semantic Web, Hawaii, May 7, 2002*, volume 55 of *CEUR Workshop Proceedings*.

Middleton, S.; Roure, D. D.; and N.R.Shadbolt. 2001. Capturing knowledge of user preferences: ontologies on recommender systems. In *In Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001)*.

Middleton, S. E.; Roure, D. D.; and Shadbolt, N. R. 2003. Ontology-based recommender systems. In Staab, S., and Studer, R., eds., *Handbook on Ontologies*. Springer.

Montaner, M.; Lopez, B.; and De La Rosa, J. L. 2003. A taxonomy of recommender agents on the internet. *Artif. Intell. Rev.* 19(4):285–330.

Nejdl, W., et al. 2003. Super-peer-based routing and clustering strategies for rdf-based peer-to-peer networks. In *Proceedings of the Twelfth International World Wide Web Conference (WWW 2003)*.

Nejdl, W.; Wolf, B.; Qu, C.; Decker, S.; Sintek, M.; Naeve, A.; Nilsson, M.; Palmér, M.; and Risch, T. 2002. Edutella: A P2P networking infrastructure based on rdf. In *Proceedings to the Eleventh International World Wide Web Conference*.

Oram, A., ed. 2001. *Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology*. Sebastopol (CA): OReilly.

Rada, R.; Mili, H.; Bicknell, E.; and Blettner, M. 1989. Development and application of a metric on semantic nets. In *IEEE Transactions on Systems, Man and Cybernetics*, 17–30.

Schein, A.; Popescul, A.; Ungar, L.; and Pennock, D. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Schwarzkopf, E. 2004. Enhancing the interaction with information portals. In *Intelligent User Interfaces 2004*, 322–324.

Tatarinov, I.; Ives, Z.; Madhavan, J.; Halevy, A.; Suciu, D.; Dalvi, N.; Dong, X.; Kadiyska, Y.; Miklau, G.; and Mork, P. 2003. The piazza peer data management project. *SIGMOD Record* 32(3).

Tempich, C.; Staab, S.; and Wranik, A. 2004. REMINDIN': Semantic query routing in peer-to-peer networks based on social metaphors. In *Proceedings of the 13th Int. World Wide Web Conference, WWW 2004*.