Bibster - A Semantics-Based Bibliographic Peer-to-Peer System

Jeen Broekstra[3], Marc Ehrig[1], Peter Haase[1], Frank van Harmelen[2], Maarten Menken[1], Peter Mika[2], Björn Schnizler[1], Ronny Siebes[2]

[1]Institute AIFB, University of Karlsruhe, Karlsruhe, Germany
{ehrig, haase}@aifb.uni-karlsruhe.de, schnizler@iw.uni-karlsruhe.de
[2]Vrije Universiteit Amsterdam, The Netherlands
{frankh, mrmenken, pmika, ronny}@cs.vu.nl
[3]Aduna, Amersfoort, The Netherlands
jeen@aduna.biz

Abstract. This paper describes the design and implementation of Bibster, a Peer-to-Peer system for exchanging bibliographic data among Computer Science researchers. Bibster exploits ontologies in data-storage, query formulation, query-routing and answer presentation: When bibliographic entries are made available for use in Bibster, they are structured and classified according to two different ontologies. This ontological structure is then exploited to help users formulate their queries. Subsequently, the ontologies are used to improve query routing across the Peer-to-Peer network. Finally, the ontologies are used to post-process the returned answers in order to do duplicate detection. The paper describes each of these ontology-based aspects of Bibster. Bibster is fully implemented on top of the JXTA platform, and is about to be rolled out for field testing.

1 Introduction

The advantages of Peer-to-Peer architectures over centralized approaches have been well advertised, and to some extent realized in existing applications: no centralized server (thus avoiding a bottleneck for both computational performance and information update), robustness against failure of any single component, scalability both in data-volumes and the number of connected parties.

However, besides being the solution to many problems, the large degree of distribution of Peer-to-Peer systems is also the cause of a number of new problems: the lack of a single coherent schema for organizing information sources across the Peer-to-Peer network hampers the formulation of search queries, duplication of information across the network results in many duplicate answers to a single query, and answers to a single query often require the integration of information residing at different, independent and uncoordinated peers [1]. Finally, query routing and network topology (which peers to connect to, and which peers to send/forward queries to) are significant problems.

The research community has recently turned to the use of semantics in Peer-to-Peer networks to alleviate these problems [2], [3], [4]. The use of semantic descriptions of datasources stored by peers and indeed of semantic descriptions

of peers themselves is claimed to help in formulating queries in such a way that they can be understood by other peers, in merging the answers received from different peers, and in directing queries across the network. In particular, the use of ontologies and of Semantic Web technologies in general has been identified as promising for Peer-to-Peer systems.

This paper describes the Bibster system[1], an application of the use of semantics in Peer-to-Peer systems. 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.

The following characteristics make this domain an interesting use case for a semantics-based Peer-to-Peer system:

- 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 DBLP[2]. Furthermore, any such centralized resource will only cover a limited scientific community. For example DBLP covers a lot of Artificial Intelligence, but almost no Knowledge Management, whereas a lot of work is being done in the overlap of these two fields.
- The use of Semantic Web technology is crucial in this setting. Although a small common-core ontology of bibliographic information exists (title, author/editor, etc.), much of this information is very volatile and users define arbitrary add-ons, for example to include URLs of publications, to include abstracts, private comments, etc.

The bibliographic domain of the Bibster system has a number of characteristics that will determine some of our design decisions:

- Researchers will want to search for bibliographic entries using simple keyword searches, but also more advanced, semantic searches, e.g. for publications of a special type, with specific attribute values, or about a certain topic or related topics.
- Researchers may want to query a single specific peer (e.g. a server with all bibliographic metadata from a specific conference), a specific set of peers (e.g. all peers that are known to have information on a given topic), or the entire network of peers (to obtain the maximal recall at the price of low precision)
- Researchers will want to integrate results of a query into a local knowledge base for future use. Such data may in turn be used to answer queries by other peers. They may also be interested in updating items that are already locally stored with additional information about these items obtained from other peers.

[1] `http://bibster.semanticweb.org/`
[2] `http://www.informatik.uni-trier.de/~ley/db/`

– As will be discussed in later sections, the ontologies that will be used in Bibster are "lightweight": simple taxonomies of terms without much further formalized meaning would seem to be appropriate for this domain.

In this paper we will describe the design of the Bibster system (section 2), and emphasize the semantic components and their use: semantic extraction of bibliographic metadata in section 3, semantic querying in section 4, peer selection using semantic topologies in section 5, and semantic duplicate detection in section 6. Furthermore, Bibster has been fully implemented and tested, and will be rolled out to a user community in the coming months, for which we describe the evaluation plan in section 7.

2 The Bibster System

The Bibster system has been implemented as an instance of the SWAP system architecture as introduced in [2]. 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.

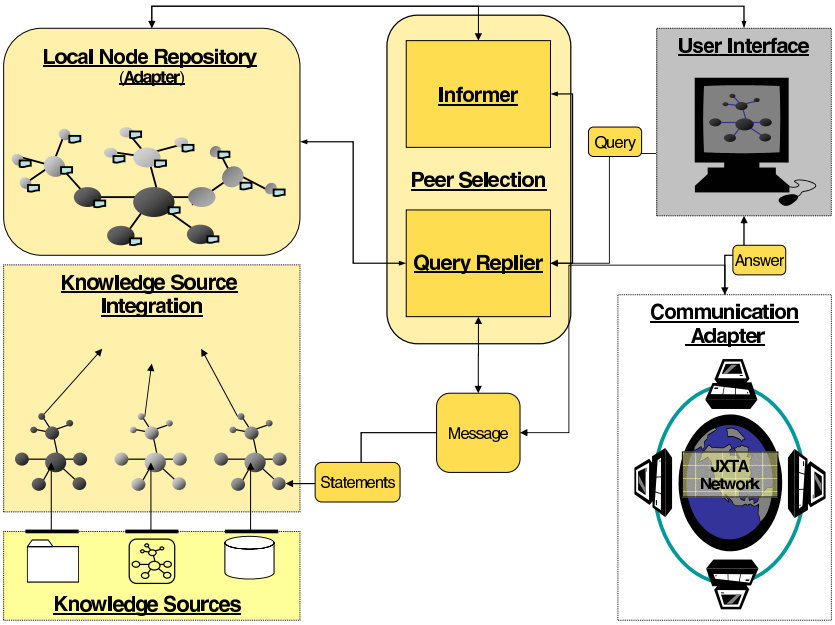

Fig. 1. SWAP system architecture

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

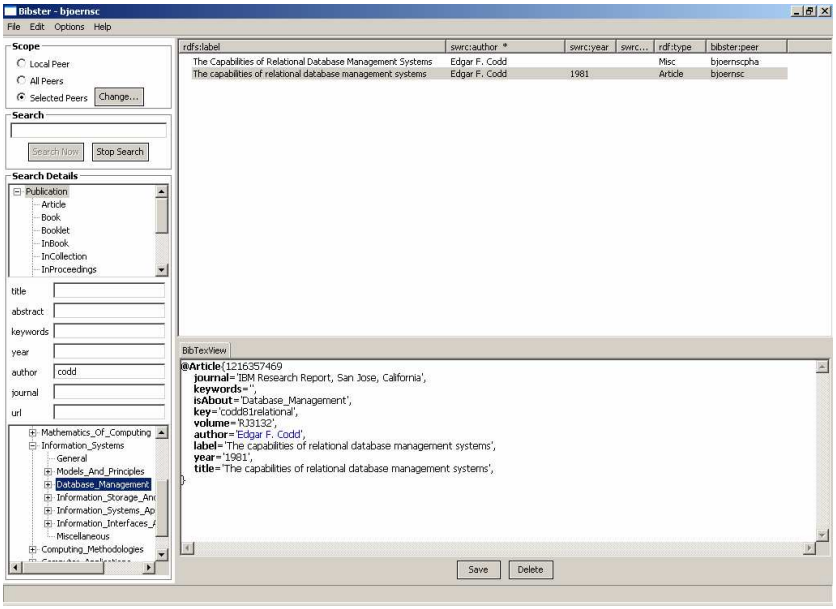

Fig. 2. User interface for the Bibster application

communication details from the rest of the system. In the specific implementation of the Bibster system we use JXTA as the communication platform.

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. In section 3 we describe the process of semantic extraction from BibTeX files. In section 6 we explain how the knowledge of local and remote sources can be merged, i.e. how duplicate query results are detected.

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 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 [5]. The query language SeRQL is used to formulate semantic queries against the Local Node Repository, as described in section 4.

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

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. This process is described in detail in section 5.

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. The decision to which peers a query should be sent is made based on the knowledge about the expertise of other peers.

User Interface The user interface, as shown in figure 2 allows the user to 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.

3 Semantic Extraction of Bibliographic Metadata

Large amounts of bibliographic metadata are stored in BibTeX files. Many researchers have accumulated extensive collections of BibTeX files for their bibliographic references. However these files are semi-structured and thus single attributes may be missing or may not be interpreted correctly [6]. Another problem is that there are no well-defined interfaces for the exchange of standard BibTeX files.

For interchanging bibliographic data in a semantics-based Peer-to-Peer network it has to be represented in a structured and formal way. The usage of standardized representations is decisive for sharing knowledge with other peers.

BibToOnto[3] is a component of Bibster for extracting explicit knowledge of bibliographic items. The tool was developed specifically for the Bibster project. Plain BibTeX will be transformed into an ontology based knowledge representation. This transformation is used to give meaning to the information structures that are to be exchanged between peers.

The target ontology is the Semantic Web Research Community Ontology (SWRC[4]), which models among others a research community, its researchers, topics, publications, tools, and properties between them [7]. The SWRC ontology defines a shared and common domain theory which helps users and machines to communicate concisely and supports exchange of semantics.

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

BibToOnto automatically classifies bibliographic entries according to the ACM topic hierarchy[5], using a simple keyword based approach. Additionally, it is possible to reclassify the entries manually in the user interface of Bibster.

The ACM topic hierarchy has become a standard schema for describing and categorizing computer science literature. It covers 1287 topics of the computer science domain. In addition to the sub- and supertopic relations, it also provides information about related topics.

The following example shows a transformation of a BibTeX entry to a SWRC ontology based item. The result is represented as an RDF graph in figure 3.

Example 1.
```
@ARTICLE{codd81relational,
  author = {Edgar F. Codd},
  title = {The capabilities of relational database management systems},
  journal = {IBM Research Report, San Jose, California},
  volume = {RJ3132},
  year = {1981}
}
```

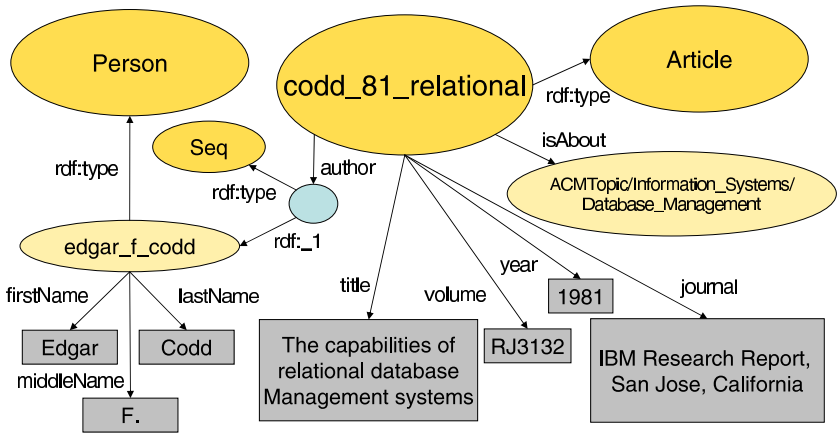

Fig. 3. SWRC Sample Metadata

Authors and editors are represented as instances of the `swrc:Person` class. They can be identified by a unique URI[6].

The publication itself is instantiated as `swrc:Article` which is a subclass of `swrc:Publication`. The order of authors is guaranteed by the use of RDF sequences. The ACM topics corresponding to the publications are represented with the `swrc:isAbout` properties. In this example, the associated topic is `Database Management`.

[5] http://www.acm.org/class/1998/

[6] For better readabilty we used a concatenation of the author name and the title of the publication as a URI in this example. In the Bibster system however we calculate hash codes over all attribute values to guarantee the uniqueness of URIs.

4 Semantic Querying

Each peer node in the Bibster system has a local RDF repository, in which both local knowledge and knowledge obtained from other peers is stored. A user interface allows users to edit, browse and query this knowledge. The de-facto standard query language in the system is SeRQL [8].

SeRQL (Sesame RDF Query Language, pronounced "circle") is an RDF(S) query language that was developed in the context of the SWAP project to address practical requirements from the Sesame user community[7] that were not sufficiently met by other query languages.

4.1 SeRQL design principles and requirements

Within the context of the SWAP project and the Bibster system, several requirements on RDF querying are of particular importance:

1. A convenient yet powerful path expression syntax for navigating RDF graphs
2. Functionality for navigating the class/property hierarchy
3. Schema awareness
4. Program manipulation
5. Functionality to deal with *optional* values, properties which may or may not be present in the data for a particular resource

In SeRQL, all of these requirements are met with a diverse set of language features. Without illustrating the complete spectrum of functionality[8], we briefly show how SeRQL queries are composed, and how tasks in the Bibster system are performed using SeRQL.

SeRQL uses a `select-from-where` or `construct-from-where` filter, where the `select` or `construct` clauses specify projections, the `from` clause specifies a graph match template (by means of path expressions), and the `where` clause allows the definition of additional boolean constraints on matched values in the path expressions.

Path expressions in SeRQL are specified as a chain of nodes and edges in the RDF graph: `{s} p {o}` is a general path expression that matches any statement. Each node is denoted with curly brackets, so in the above expression `s` and `o` match nodes, while `p` matches edges in the graph.

One of the requirements was that the query language should be schema aware. SeRQL enables this by specifying a mapping to the formal model of RDF and RDF-S, and using this formal mapping to specify the meaning of path expressions where path labels have predefined semantics. For example, `<rdfs:subClassOf>` is interpreted as a reflexive transitive relation, upward inheritance of instances is interpreted (through the `<rdf:type>` relation), etc. In [8] a complete mapping to the RDF semantics is provided.

Program manipulation of queries is an important aspect in the Bibster system. For example, the user interface automatically transforms the user input into SeRQL queries on the underlying repository, and also transforms the SeRQL result to a representation in the UI for the user. This requires not only that the

[7] See `http://www.openrdf.org/`

[8] See `http://www.openrdf.org/doc/SeRQLmanual.html` for a complete overview

query language syntax is simple to parse and write yet unambiguous, but also that the query result is returned in a format that can be easily processed automatically by the client. SeRQL, in the case of construct-queries, returns RDF graphs in the form of RDF/XML documents.

Another requirement on the query language states that it should be compositional. In the case of SeRQL, compositionality means that the result of a query should be an RDF graph (this is achieved with the construct-clause mentioned earlier). The effect of this is that the query language functions as a *transformation language* on RDF graphs. This notion is of particular interest for a system such as Bibster, or more generally, the SWAP architecture, since it allows peers to easily integrate obtained results from queries into their own knowledge base.

4.2 Querying in the Bibster system

In our scenario, a researcher is looking for journal articles written by the author Codd about database management. The user speficies his search request through the user interface as shown in figure 2. Internally, this request is formulated as a SeRQL query that looks as follows:

Example 2.

```
construct distinct
    {s} prop {val};
        <rdf:type> {t};
        <swrc:author> {x} <rdf:type> {<rdf:Seq>};
                          <rdfs:member> {author} prop_author {val_author}
from
    {s} <serql:directType> {t};
        <rdf:type> {<swrc:Article>};
        prop {val};
        <swrc:isAbout> {<acm:ACMTopic/Information_Systems/Database_Management>};
        <swrc:author> {x} <rdfs:member> {} <swrc:lastName> {lname},
        [{x} <rdfs:member> {author} prop_author {val_author} ]
where prop != <rdf:type> and lname like "Codd"
using namespace
    swrc = <!http://www.semanticweb.org/ontologies/swrc-onto-2001-12-11.daml#>,
    acm =  <!http://daml.umbc.edu/ontologies/classification#>
```

Compare the structure of the from-clause to the representation of the RDF graph given in figure 3. The from-clause retrieves not only the identifier for the particular journal entry ("codd_81_relational", matched by `s`), but also the graph structure surrounding it, which essentially gives the entry its meaning: the name of the author, the type of publication, the year it was published, the number of pages, etc. Also, if the first and middle names of an author are known, the query retrieves those (but it does not fail if these are not known).

The use of schema-awareness is evident in the use of typing information on `s`: not only must it be of type `swrc:Article`, we also retrieve its *specific* (or *direct*) type. Compositionality plays a role as well: a graph transformation is used to create a query result that can be easily processed to be given back to the user through the GUI.

5 Expertise Based Peer Selection

The scalability of a Peer-to-Peer network is essentially determined by the way how queries are propagated in the network. Peer-to-Peer networks that broadcast

all queries to all peers do not scale – intelligent query routing and network topologies are required to be able to route queries to a relevant subset of peers that are able to answer the queries.

Modern routing protocols like Chord [9], CAN [10] and Pastry [11] allow for sophisticated routing based on distributed indices. More recently, in the Semantic Web context, schema based Peer-to-Peer networks such as the one described in [12] have emerged that are based on complex, extendable and semantic descriptions of resources instead of fixed and limited ones. They allow for complex query facilities against these metadata instead of simple keyword-based queries. Another semantics-based approach is pSearch [13], a decentralized non-flooding P2P information retrieval system. pSearch distributes document indices through the P2P network based on document semantics generated by Latent Semantic Indexing (LSI). The search cost (in terms of different nodes searched and data transmitted) for a given query is thereby reduced, since the indices of semantically related documents are likely to be co-located in the network.

In this section we provide an overview of the model of expertise based peer selection as proposed in [14] and how it is used in the Bibster system. We also show results of simulation experiments that we performed to evaluate the model.

5.1 Model of Expertise Based Peer Selection

In the model that we propose, peers use a shared ontology to advertise their expertise in the Peer-to-Peer network. The knowledge about the expertise of other peers forms a semantic topology, independent of the underlying network topology. If the peer receives a query, it can decide to forward it to peers about which it knows that their expertise is similar to the subject of the query. The advantage of this approach is that queries will not be forwarded to all or a random set of known peers, but only to those that have a good chance of answering it. The peer selection is based on matching the subject of a query and the expertise according to their semantic similarity.

In the following, we first introduce a model to semantically describe the expertise of peers and how peers promote their expertise as advertisement messages in the network. Second, we describe how the received advertisements allow a peer to select other peers for a given query based on a semantic matching of query subjects against expertise descriptions. The third part describes how a *semantic topology* is formed by advertising expertise.

Semantic Description of Expertise

Peers The Peer-to-Peer network consists of a set of peers P. Every peer $p \in P$ has a knowledge base that contains the knowledge that it wants to share. In Bibster, the knowledge base is the Local Node Repository, which stores the bibliographic metadata.

Common Ontology The peers share an ontology O, which provides a common conceptualization of their domain. The ontology is used for describing the expertise of peers and the subject of queries. In our case, O is the ACM topic hierarchy that contains a set of topics T.

Expertise An expertise description $e \in E$ is an abstract, semantic description of the knowledge base of a peer based on the common ontology O. The ACM ontology is the basis for our expertise model. Expertise E is defined as $E \subseteq 2^T$, where each $e \in E$ denotes a set of ACM topics, for which a peer provides classified instances.

Advertisements Advertisements $A \subseteq P \times E$ are used to promote descriptions of the expertise of peers in the network. An advertisement $a \in A$ associates a peer p with an expertise e. Peers decide autonomously, without central control, whom to promote advertisements to and which advertisements to accept. This decision can be based on the semantic similarity between expertise descriptions.

Matching and Peer Selection

Queries Queries $q \in Q$ are posed by a user and are evaluated against the knowledge bases of the peers. First a peer evaluates the query against its local knowledge base and then decides which peers the query should be forwarded to. In Bibster, we use the SeRQL query language, as presented in previous section.

Subjects A subject $s \in S$ is an abstraction of a given query q expressed in terms of the common ontology. The subject can be seen as a complement to an expertise description, as it specifies the required expertise to answer the query. In our scenario, each s is the set of ACM topics that are referenced in the query. Thus $s \subseteq T$. For example, the extracted subject of the query in example 2 would be *Information Systems/Database Management*.

Similarity Function The similarity function $Sim : S \times E \mapsto [0, 1]$ yields the semantic similarity between a subject $s \in S$ and an expertise description $e \in E$. An increasing value indicates increasing similarity. In this scenario, the similarity function Sim_{Topics} is based on the idea that topics which are close according to their positions in the topic hierarchy are more similar than topics that have a larger distance. For example, an expert on the ACM topic *Information Systems/Information Storage and Retrieval* has a higher chance of giving a correct answer on a query about *Information Systems/Database Management* than an expert on a less similar topic like *Hardware/Memory Structures*.

To be able to define the similarity of a peer's expertise and a query subject, which are both represented as a set of topics, we first define the similarity for individual topics. [15] have compared different similarity measures and have shown that for measuring the similarity between concepts in a hierarchical structured semantic network, like the ACM topic hierarchy, the following similarity measure yields the best results:

$$sim_{Topic}(t_1, t_2) = \begin{cases} e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} & \text{if } t_1 \neq t_2, \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

Here l is the length of the shortest path between topic t_1 and t_2 in the graph spanned by the *SubTopic* relation. h is the level in the tree of the direct common subsumer from t_1 and t_2. $\alpha \geq 0$ and $\beta \geq 0$ are parameters scaling the contribution of shortest path length l and depth h, respectively. Based on their benchmark data set, the optimal values are: $\alpha = 0.2$, $\beta = 0.6$.

Now that we have a function for calculating the similarity between two individual topics, we define Sim_{Topics} as:

$$Sim_{Topics}(s, e) = \frac{1}{|s|} \sum_{t_i \in s} \max_{t_j \in e} sim_{Topic}(t_i, t_j) \quad (2)$$

With this function we iterate over all topics of the subject and average their similarities with the most similar topic of the expertise.

Peer Selection Algorithm The peer selection algorithm returns a ranked set of peers, where the rank value is equal to the similarity value provided by the similarity function. Therefore, peers that have an expertise more similar to that of the subject of the query will have a higher rank. From this set of ranked peers one can, for example, select the best n peers, or all peers whose rank value is above a certain threshold. In the Bibster system we selects the best n peers, where n can be specified.

Semantic Topology

The knowledge of the peers about the expertise of other peers is the basis for a semantic topology. Here it is important to state that this semantic topology is independent of the underlying network topology. At this point, we do not make any assumptions about the properties of the topology on the network layer.
The semantic topology can be described by the following relation:

$Knows \subseteq P \times P$, where $Knows(p_1, p_2)$ means that p_1 knows about the expertise of 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 semantic topology in combination with the expertise based peer selection is the basis for intelligent query routing.

5.2 Results of Simulation Experiments

The proposed model for expertise based peer selection has been evaluated in an experimental simulation environment [14]. As a data set we used a subset of the DBLP database consisting of 126247 bibliographic entries which we were able to classify using the simple classification scheme described before. We simulated two different document distributions:

- *Topic Distribution* with 1287 peers, where each peer is an expert on one of the 1287 ACM topics and contains all the bibliographic entries for that topic, and the
- *Proceedings Distribution* with 2335 peers, where each peer contains the bibliographic entries of a certain journal or conference proceedings.

The evaluation criteria of the experiments are mainly based on those presented in [16]: We have considered precision and recall both on the document level (query answering) and peer level (peer selection), as well as the number of messages generated per query. The results of our experiments can be summarized as follows:

- **Expertise based selection:** The proposed approach of expertise based peer selection yields better results than a naive approach based on random selection. The higher precision of the expertise based selection results in a higher recall of peers and documents, while reducing the number of messages per query.
- **Ontology based matching:** Using a shared ontology with a metric for semantic similarity improves the recall rate of the system compared with an approach that relies on exact matches, such as a simple keyword based approach.
- **Semantic topology:** The performance of the system can be improved further, if the semantic topology is built according to the semantic similarity of the expertises of the peers. This can be realized, for example, by accepting advertisements that are semantically similar to the own expertise.
- **The "Perfect" topology:** Perfect results in terms of precision and recall can be achieved, if the semantic topology coincides with a distribution of the documents according to the expertise model. This is for example the case in the topic distribution, if each peer "knows" the peers that are experts on the super- and subtopics of its own expertise.

We will validate the results from our simulation experiments in the field experiment as the described in section 7 based on the same evaluation scheme.

6 Semantic Duplicate Detection

When querying the Bibster network we expect to receive a large number of results with a potentially high number of duplicates, even if the query itself is very restrictive already. The large number of results is due to the fact that we do not have a centralized repository but a Peer-to-Peer network. Furthermore, as the metadata is created in a distributed and decentralized manner, the representation of the metadata is very heterogeneous and possibly even inconsistent. To enable an efficient and easily usable system it is necessary to filter these duplicates. Specifically, we do not want to confront the user with a list of all individual results. Therefore we present query results grouped by semantic duplicates.

6.1 Process

Our proposed model for the detection of duplicates is based on the notion of semantic similarity. Duplicates are bibliographic entries which refer to the same publication or person in the real world, but are modelled as different resources. We will now present a definition of a similarity function, based on which we define the duplicate relation and its application for grouping query results.

Similarity Function As described in the section 4, query results are represented in RDF. 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 [17]. This function is based on different features of the respective resources. Besides individual functions, our approach applies an aggregation function to achieve an overall similarity result.

The Duplicate Relation As *duplicates* we consider those pairs of resources whose similarity is larger than a certain threshold $t \in [0..1]$:

$$D_t := \{(x, y)|sim(x, y) \geq t\}$$

If we assume that the duplicate relation is transitive, we can define the transitive closure as:

$$TC(D_t) := \{(x, z)|(x, y) \in D_t \wedge (y, z) \in D_t\}$$

This transitive closure essentially represents clusters of semantically similar resources, which are presented as such.

Resource Merging Instead of presenting all resources of the query result, duplicates are visualized as one, merged, resource. These merged resources consist of a union of properties of the individuals identified as duplicates. In the case of conflicting or inconsistent property values, we apply heuristics (e.g. to select the more complete value) for the merging of resources.

6.2 Similarity Methods

We now show instantiations of similarity methods for persons, organizations and publications as defined in the SWRC ontology.

Features Each entity type is compared through specific features. For persons we use the first, middle, and last names. For organizations we rely solely on the organization name. And 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 each of the features we use different specific functions, which can be grouped as follows. For each of the levels we will provide representative example functions.

Data Value Level: The data value level focuses on comparisons of data values, which in RDF are represented as typed literals.
For example, to determine the similarity of data values d_1, d_2 of type string (e.g. to compare the last names of persons) we use the *syntactic similarity* sim_{syn} of [18]. It is inverse to the edit distance (ed) of [19], which basically determines how many atomic actions as character addition or deletion are required to transform one string into the other one.

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

Analogously, one can define similarity functions for other datatypes.

Graph Structure Level On this level we make use of the graph structure, specifically we check how resources are related with each other. For example, 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:

$$sim_{set}(E,F) := \frac{1}{|E|+|F|} \cdot \left(\sum_{e \in E} max_{f \in F} sim(e,f) + \sum_{f \in F} max_{e \in E} sim(f,e) \right)$$

Ontology Level The ontology level enhances the simple graph structure level through ontology specific characteristics such as the taxonomy.
For example, to determine the similarity of two topics t_1 and t_2 of a topic hierarchy we apply the following similarity function, which was already explained in the previous section:

$$sim_{Topic}(t_1,t_2) = \begin{cases} e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} & \text{if } t_1 \neq t_2, \\ 1 & \text{otherwise} \end{cases}$$

Again, the parameters are set to $\alpha = 0.2$, $\beta = 0.6$.

Domain Specific Knowledge Applying background knowledge about a specfic domain, we can define more appropriate similarity functions.
For example, 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[9]).
Instead of using a generic similarity function, we can thus define:

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

Aggregated Similarity Function From the variety of individual similarity functions, an overall value is obtained with an aggregated similarity function, using a weighted average over the individual functions:

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

with w_k being the weight for a specific function sim_k. For the Bibster scenario, the weights have been assigned based on experiments with sample data. 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.

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

6.3 Example

We refer to the example extracted from BibTeX in figure 3; specifically, we are interested in the publication (p_1). We further assume to have another publication entry (p_2) as shown in figure 4. Please note the differences in representation for the two publications, as they are characteristic of BibTeX: There are syntactic differences (lower case vs. upper case in the title), abbreviations (in journal), and subjective classifications (for the ACM topic and type: Misc. vs. Article) and incomplete or missing fields (year).

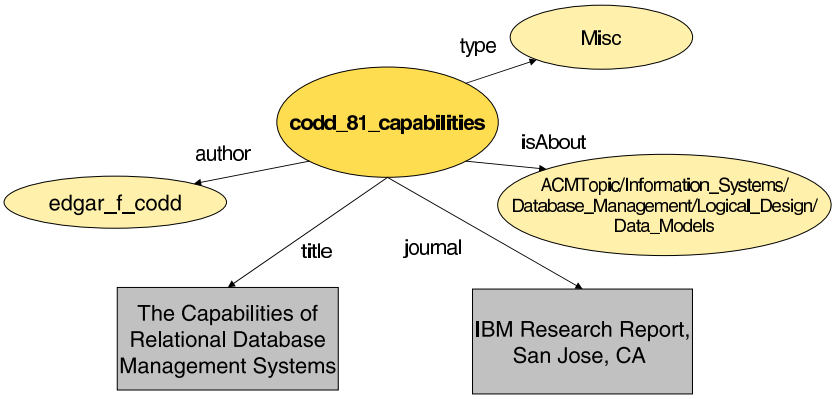

Fig. 4. Second SWRC example

When comparing the two example publications applying the similarity functions from above we obtain:

$$
\begin{aligned}
sim_{title}(p_1,p_2) &= sim_{syn} && (\textit{“The capabilities of relational database management systems”}, \textit{“The Capabilities of Relational Database”}) && = 0.91\\
sim_{type}(p_1,p_2) &= sim_{type} && (Article, Misc) && = 0.75\\
sim_{author}(p_1,p_2) &= sim_{set} && (\{edgar\_f\_codd\},\{edgar\_f\_codd\}) && = 1\\
sim_{journal}(p_1,p_2) &= sim_{syn} && (\textit{“IBM Research Report, San Jose, California”}, \textit{“IBM Research Report, San Jose, CA ”}) && = 0.72\\
sim_{topic}(p_1,p_2) &= sim_{topic} && (Database\_Management, Data\_Models) && = 0.56
\end{aligned}
$$

$$
\begin{aligned}
sim_{agg}(p_1,p_2) = \tfrac{1}{10+5+8+5+5} \cdot (&sim_{title}(p_1,p_2)\cdot 10 + sim_{publication}(p_1,p_2)\cdot 5\\
&+sim_{author}(p_1,p_2)\cdot 8 + sim_{booktitle}(p_1,p_2)\cdot 5\\
&+sim_{topic}(p_1,p_2)\cdot 5) = 0.82
\end{aligned}
$$

In the Bibster system we use a threshold[10] of $t = 0.8$, we therefore identify these two resources as duplicates: $D = \{(p_1,p_2)\}$. The merged resource will be identical with p_1, except for the extended topic classification (. . . /Logical_Design/Data_Models).

[10] The threshold, as the weights, has been assigned from experiments with sample data.

7 Evaluation

7.1 Evaluation Plan

The methods and the implementation presented in this paper will be evaluated by means of a study among the potential end users of the system. Although we cannot report on the results of this study at the time of writing, we present our evaluation plan in order promote principled evaluation and higher transparency in assessing Peer-to-Peer platforms and applications. The methodology for the evaluation is based on the guidelines described in the SWAP methodology [20]. Accordingly, the evaluation plan follows a top-down approach to planning.

In the following, we will describe the evaluation goals and measures in Section 7.2 and the methods for user-focused and system evaluation in Sections 7.3 and 7.4, respectively[11]. System evaluation refers to logging actions of the system and its interaction with the user, while user evaluation concerns feedback elicited directly from the users themselves. Note that user evaluation and system evaluation offer different insights and need to be combined to obtain meaningful results [16].

The case study will involve some of the targeted end users of the system, who will use the system in their daily work for a period of six months. The study will begin in April, 2004 with a core group of researchers representing a mix of research areas in Computer Science and different levels of research experience. The number of participants in this phase will be 50–60 persons. Additionally, the system will be made available publicly through the project website[12] starting from May, 2004.

7.2 Evaluation Goals and Measures

In our study, we are looking for answers to the following generic questions, which we use to generate our evaluation measures:

1. Assess if the system improves knowledge sharing and supports community awareness and formation. With respect to community awareness, we are interested if the system is able to create weak ties, i.e. "connect" scientists who have not known each other previously and were unaware of each other's work. User satisfaction is an important indicator for both of these aspects.
2. Estimate the particular benefit from ontology use in the system. Bibster improves on existing P2P systems by incorporating semantics in the peer selection process of query routing. Semantics-based peer selection increases the efficiency of the system as demonstrated within a simulation environment (see Section 5.2). The results of this experiment will be validated by using the same method in the case study.
3. Lastly, we would like to explore the usage patterns and the network structures emerging from the use of the system. Observing the emergent semantic topology and its relation to the social networks of users should enable us to create more targeted architectures and algorithms in the future.

[11] For a detailed version of the evaluation plan we refer the reader to [21].
[12] `http://bibster.semanticweb.org/`

7.3 User-Focused Evaluation

User evaluation of the bibliographic case study will consist of two post-trial questionnaires to be filled out by the end users of the system. Selected users will also be called for personal interviews to obtain more open-ended feedback.

The first questionnaire, called SUMI (Software Usability Measurement Inventory) is an industry standard for measuring software quality from the end users' point of view. SUMI is a set of 50 questions (propositions) to be answered with one of three choices: Agree, Don't know or Disagree. The second questionnaire is a specific test for assessing the user experience with certain particular features of the Bibster system. Further, this questionnaire profiles the users in terms of their technical environment, scientific background and existing social relationships with each other. Techniques from Social Network Analysis will be applied to find out how these preexisting social relationships influence the socio-technical ecosystem emerging from system use.

7.4 System Evaluation

In addition to the user evaluation we do a system evaluation, which refers to evaluation through automated data collecting, i.e. a recording and analysis of user and system activity by means of log files. The decision on what events to log is based on predefined evaluation measures, like semantic topology and knowledge exchange network.

The log files are created locally on each peer and stored in XML format to simplify future processing. Periodically the local log files are sent over the Internet to a central server. This is done automatically without user intervention. The gathered log files are then aggregated to allow overall evaluation.

8 Related Work

In the previous sections, related work on the individual aspects of semantics-based Peer-to-Peer technology has already been discussed. Therefore in this section our study of related work focuses on complete systems.

Edutella [3] is a Peer-to-Peer system based on the JXTA platform, which offers very similar base functionality as the SWAP system. The Edutella network uses the query exchange language family RDF-QEL as standardized query exchange language format which is transmitted in an RDF/XML-format. Unlike the query language SeRQL used in our system, QEL is a query exchange language format based on Datalog semantics, which is mapped to specific query languages such as SQL or RQL. [12] presents schema-based Peer-to-Peer networks and the use of super-peer based topologies for these networks, in which peers are organized in hypercubes. [22] shows 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. [23] describes 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.

P-Grid [24] is a structured, yet fully-decentralized Peer-to-Peer system based on a virtual distributed search tree. It aims at providing load-balancing and fault-tolerance, assuming that peers fail frequently and are online with low probability. P-Grid also considers updates with an update algorithm based rumor spreading.

Various systems address the issue of heterogeneity in Peer-to-Peer systems on the schema level, such as the Piazza peer data management system [25], which allows for information sharing with different schemas relying on local mappings between schemas. [2] proposes a metadata model that allows to annotate information in the Peer-to-Peer network with meta-information about origin, confidence, trust, etc. to address heterogeneity and inconsistencies also on the metadata or instance level. However, it does not provide a model for detecting duplicates, as presented in this paper.

9 Conclusion and Future Work

In this paper, we have described the design and implementation of a semantics-based Peer-to-Peer system for the exchange of bibliographic metadata between researchers. In this concluding section, we review how the use of ontologies is crucial in all the steps of Bibster: importing data, formulating queries, routing queries, and processing answers.

Firstly, the system enables users to import their own bibliographic metadata into a local repository. This bibliographic metadata is made available under a two common ontologies: the first ontology (SWRC) describes different generic aspects of bibliographic metadata (and would be valid across many different research domains), the second ontology (ACM Topic Hierarchy) describes specific categories of literature for the Computer Science domain. Bibliographic entries that are made available to Bibster by a user are automatically classified under these two ontologies. Both the ontologies and the specific bibliographic instance data are represented in RDF.

Secondly, users can send queries to other peers looking for bibliographic metadata. These queries are formulated in terms of the two ontologies: queries can concern fields like author, publication type etc. (using terms from the SWRC ontology) or queries can concern specific Computer Science terms (using the ACM Topic Hierarchy). These user-queries are translated into the RDF query language SeRQL to be answered by the different peers in the network.

Thirdly, these queries need to be routed across the peer-network, and again the ontologies play a crucial role. Queries are routed through the network depending on the expertise models of the peers. Such an expertise model describes which concepts from the ACM ontology a peer can answer queries on. A matching function determines how closely the semantic content of a query matches the expertise model of each peer. Routing is then done on the basis of this semantic ranking.

Finally, answers are returned for a query. Due to the distributed nature and potentially large size of the Peer-to-Peer network, this answer set might be very large, and contain many duplicate answers. Because of the semistructured nature of bibliographic metadata, such duplicates are often not exactly identical copies.

Again in this step, we exploit ontologies, this time to measure the semantic similarity between the different answers, and to remove apparent duplicates as identified by the similarity function.

As is clear from the above, we exploit lightweight ontologies, expressed in RDF Schema in all the crucial aspects of our system: data-organisation, query formulation, query routing and duplicate detection.

In order to measure the effectiveness of our semantics-based approach, we are planning to execute an extensive evaluation study, measuring both user-related aspects (such as user-satisfaction with interface, performance, etc.), and system-related aspects (such as average number of hops for a query, number of duplicates detected, etc.).

The Bibster system is one of the first ontology-based Peer-to-Peer systems ready for fielded deployment, which uses ontologies in all its steps. Particularly interesting will be to see how its performance will compare to related systems such as P-Grid and Edutella.

Acknowledgments

Research reported in this paper has been partially financed by the EU in the IST project SWAP (IST-2001-34103). We would like to thank our colleagues for fruitful discussions.

References

1. Oram, A., ed.: Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology. OReilly, Sebastopol (CA) (2001)
2. Broekstra, J., Ehrig, M., Haase, P., van Harmelen, F., Kampman, A., Sabou, M., Siebes, R., Staab, S., Stuckenschmidt, H., Tempich, C.: 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. (2003)
3. Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmér, M., Risch, T.: Edutella: A P2P networking infrastructure based on RDF. In: Proceedings to the Eleventh International World Wide Web Conference, Honolulu, Hawaii, USA (2002)
4. Castano, A., Ferrara, S., Montanelli, S., Pagani, E., Rossi, G.: Ontology-addressable contents in P2P networks. In: Proceedings of the WWW'03 Workshop on Semantics in Peer-to-Peer and Grid Computing. (2003)
5. J. Broekstra, A. Kampman, F.v.H.: Sesame: An architecture for storing and querying RDF data and schema information. In D. Fensel, J. Hendler, H.L., Wahlster, W., eds.: Semantics for the WWW. MIT Press (2001)
6. Abiteboul, S.: Querying semi-structured data. In: Proceedings of the Sixth International Conference on Database Theory, Springer-Verlag (1997) 1–18
7. Handschuh, S., Staab, S., Maedche, A.: CREAM - creating relational metadata with a component-based. In: Proceedings of the First International Conference on Knowledge Capture K-CAP 2001. (2001)
8. Broekstra, J., Kampman, A.: SeRQL: An RDF query and transformation language (2004) Submitted to the International Semantic Web Conference, ISWC 2004.
9. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for Internet applications. In: Proceedings of the ACM SIGCOMM '01. (2001)

10. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content-addressable network. In: Proc. of ACM SIGCOMM '01. (2001)
11. Rowstron, A., Druschel, P.: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany (2001) 329–350
12. Nejdl, W., et al.: 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), Budapest, Hungary (2003)
13. Tang, C., Xu, Z., Mahalingam, M.: pSearch: Information retrieval in structured overlays. In: ACM HotNets-I. (2002)
14. Haase, P., Siebes, R.: Peer selection in peer-to-peer networks with semantic topologies. Technical report, AIFB, University of Karlsruhe (2004) `http://www.aifb.uni-karlsruhe.de/WBS/pha/publications/haase_siebes04.pd%f`.
15. Li, Y., Bandar, Z.A., McLean, D.: An approach for measuring semantic similarity between words using multiple information sources. Transactions on Knowledge and Data Engineering **15** (2003) 871–882
16. Ehrig, M., Schmitz, C., Staab, S., Tane, J., Tempich, C.: Towards evaluation of peer-to-peer-based distributed knowledge management systems. In van Elst, L., Dignum, V., Abecker, A., eds.: Proceedings of the AAAI Spring Symposium "Agent-Mediated Knowledge Management (AMKM-2003)". Springer LNAI, Stanford, California, Stanford University (2003)
17. Bisson, G.: Why and how to define a similarity measure for object based representation systems. Towards Very Large Knowledge Bases (1995) 236–246
18. Maedche, A.: Comparing ontologies - similarity measures and a comparison study. Technical report, Forschungszentrum Informatik, Karlsruhe, Germany (2001)
19. Levenshtein, I.V.: Binary codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory (1966)
20. Mika, P.: HOPE: Harnessing Ontologies and Peer-to-peer. SWAP project deliverable D1.5, Vrije Universiteit, Amsterdam (2003)
21. Siebes, R., Mika, P., Menken, M., Haase, P.: Evaluation plan. SWAP Project Deliverable D10.2, Vrije Universiteit, Amsterdam and University of Karlsruhe (2003)
22. Löser, A., et al.: Efficient data store discovery in a scientific P2P network. In Ashish, N., Goble, C., eds.: Proc. of the WS on Semantic Web Technologies for Searching and Retrieving Scientific Data. CEUR WS 83 (2003) Colocated with the Second International Semantic Web Conference (ISWC-03) `http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-83/`.
23. Ahlborn, B., Nejdl, W., Siberski, W.: OAI-P2P: A peer-to-peer network for open archives. In: Workshop on Distributed Computing Architectures for Digital Libraries - ICPP2002. (2002)
24. Aberer, K., Mauroux, P.C., Datta, A., Despotovic, Z., Hauswirth, M., Punceva, M., Schmidt, R.: P-Grid: a self-organizing structured p2p system. ACM SIGMOD Record **32** (2003) 29–33
25. Tatarinov, I., Ives, Z., Madhavan, J., Halevy, A., Suciu, D., Dalvi, N., Dong, X., Kadiyska, Y., Miklau, G., Mork, P.: The piazza peer data management project. SIGMOD Record **32** (2003) `http://www.acm.org/sigmod/record/issues/0309/B9.tatrinov.pdf`.