BibSonomy: A Social Bookmark and Publication Sharing System

Andreas Hotho,[1] Robert Jäschke,[1,2] Christoph Schmitz,[1] Gerd Stumme[1,2]

[1] Knowledge & Data Engineering Group, Department of Mathematics and Computer Science, University of Kassel, Wilhelmshöher Allee 73, D–34121 Kassel, Germany
http://www.kde.cs.uni-kassel.de
[2] Research Center L3S, Expo Plaza 1, D–30539 Hannover, Germany
http://www.l3s.de

Abstract. Social bookmark tools are rapidly emerging on the Web. In such systems users are setting up lightweight conceptual structures called folksonomies. The reason for their immediate success is the fact that no specific skills are needed for participating. In this paper we specify a formal model for folksonomies and briefly describe our own system BibSonomy, which allows for sharing both bookmarks and publication references in a kind of personal library.

1 Introduction

Complementing the Semantic Web effort, a new breed of so-called "Web 2.0" applications is currently emerging on the Web. These include user-centric publishing and knowledge management platforms like Wikis, Blogs, and social resource sharing tools.

These tools, such as Flickr[3] or del.icio.us,[4] have acquired large numbers of users within less than two years.[5] The reason for their immediate success is the fact that no specific skills are needed for participating, and that these tools yield immediate benefit for each individual user (e.g. organizing ones bookmarks in a browser-independent, persistent fashion) without too much overhead. Large numbers of users have created huge amounts of information within a very short period of time. The widespread use of these systems shows clearly that folksonomy-based approaches are able to overcome the knowledge acquisition bottleneck, which was a serious handicap for many knowledge-based systems in the past.

Social resource sharing systems all use the same kind of lightweight knowledge representation, called *folksonomy*. The word "folksonomy" is a blend of the words "taxonomy" and "folk", and stands for conceptual structures created by the people. Folksonomies are thus a bottom-up complement to more formalized Semantic Web technologies, as they rely on *emergent semantics* [24, 25] which result from the converging use of the same vocabulary. The main difference to "classical" ontology engineering approaches is their aim to respect to the largest possible extent the request of non-expert users not to be bothered with any formal modeling overhead. Intelligent techniques may well be inside the system, but should be hidden from the user.

[3] `http://www.flickr.com` [4] `http://del.icio.us` [5] From discussions on the del.icio.us mailing list, one can approximate the number of users on del.icio.us to be more than three hundred thousand.

This paper is organized as follows. We will shortly introduce and review existing resource sharing systems. In Section 3 we will introduce a formal model and in Section 4 we will present BibSonomy and discuss the functionality and architecture of this system.

2 Social Resource Sharing and Folksonomies

Social resource sharing systems are web-based systems that allow users to upload their resources, and to label them with arbitrary words, so-called *tags*. The systems can be distinguished according to what kind of resources are supported. Flickr, for instance, allows the sharing of photos, del.icio.us the sharing of bookmarks, CiteULike[6] and Connotea[7] the sharing of bibliographic references, and 43Things[8] even the sharing of goals in private life. Our own system, *BibSonomy*,[9] allows sharing bookmarks and BIBTEX entries simultaneously.

In their core, these systems are all very similar. Once a user is logged in, he can add a resource to the system, and assign arbitrary tags to it. The collection of all his assignments is his *personomy*, the collection of all personomies constitutes the *folksonomy*. The user can explore his personomy, as well as the personomies of the other users, in all dimensions: for a given user one can see all resources he has uploaded, together with the tags he has assigned to them (see Figure 1 on page 4); when clicking on a resource one sees which other users have uploaded this resource and how they tagged it; and when clicking on a tag one sees who assigned it to which resources.

The systems allow for additional functionality. For instance, one can copy a resource from another user, and label it with one's own tags. Overall, these systems provide a very intuitive navigation through the data.

3 A Formal Model for Folksonomies

A folksonomy describes the users, resources, and tags, and the user-based assignment of tags to resources. We present here a formal definition of folksonomies, which is also underlying our BibSonomy system.

Definition 1. *A* folksonomy *is a tuple* $\mathbb{F} := (U, T, R, Y, \prec)$ *where*

- U, T, *and* R *are finite sets, whose elements are called* users, tags *and* resources, *resp.,*
- Y *is a ternary relation between them, i. e.,* $Y \subseteq U \times T \times R$*, whose elements are called tag assignments (*tas *for short), and*
- $\prec$ *is a user-specific subtag/supertag-relation, i. e.,* $\prec \subseteq U \times T \times T$*, called* is-a relation.

Definition 2. *The* personomy $\mathbb{P}_u$ *of a given user* $u \in U$ *is the restriction of* $\mathbb{F}$ *to* u*, i. e.,* $\mathbb{P}_u := (T_u, R_u, I_u, \prec_u)$ *with* $I_u := \{(t, r) \in T \times R \mid (u, t, r) \in Y\}$*,* $T_u := \pi_1(I_u)$*,* $R_u := \pi_2(I_u)$*, and* $\prec_u := \{(t_1, t_2) \in T \times T \mid (u, t_1, t_2) \in \prec\}$*, where* π_i *denotes the projection on the* i*th dimension.*

[6] http://www.citeulike.org [7] http://www.connotea.org
[8] http://www.43things.com [9] http://www.bibsonomy.org

Users are typically described by their user ID, and tags may be arbitrary strings. What is considered as a resource depends on the type of system. For instance, in del.icio.us, the resources are URLs, in flickr, the resources are pictures, and in BibSonomy they are either URLs or publication entries. In BibSonomy each resource is represented by a hash value which is described further in Section 4.4.

Definition 3. *For convenience we also define the set P of all* posts *as*

$$P := \{(u, S, r) \mid u \in U, r \in R, S = \text{tags}(u, r)\}$$

where, for all $u \in U$ and $r \in R$: $\text{tags}(u, r) := \{t \in T \mid (u, t, r) \in Y\}$ *denotes all tags the user u assigned to the resource r.*

If we disregard the is-a relation, we can simply note a folksonomy as a quadruple $\mathbb{F} := (U, T, R, Y)$. This structure is known in Formal Concept Analysis [27, 6] as a *triadic context* [13, 26]. An equivalent view on this structure is that of a tripartite (undirected) hypergraph $G = (V, E)$, where $V = U \dot\cup T \dot\cup R$ is the set of nodes, and $E = \{\{u, t, r\} \mid (u, t, r) \in Y\}$ is the set of hyperedges.

In a typical folksonomy system every tag assignment is connected with several other properties like date, group or resource type. For sake of simplicity we disregard these properties for the rest of the paper, unless stated otherwise.

4 BibSonomy— A Folksonomy-Based Social Bookmark System

This section briefly describes the BibSonomy system[10] developed by our group. After an introduction to the user interface, semantics and architecture of BibSonomy, we explain further features as well as future enhancements. BibSonomy allows a user to share bookmarks (i.e., URLs) as well as publication references. The data model of the publication part is based on BibTeX [18], a popular literature management system for LaTeX [12].

4.1 User Interface

A typical list of posts is depicted in Figure 1 which shows bookmark and publication posts containing the tag *web*. The page is divided into four parts: the header (showing information such as the current page and path, navigation links and search boxes), two lists of posts – one for bookmarks and one for publications – each sorted by date in descending order, and a list of tags related to the posts. This scheme holds for all pages showing posts and allows for navigation in all dimensions of the folksonomy. The semantics of those pages is explained in Section 4.2.

A detailed view of one bookmark post from the list in Figure 1 can be seen in Figure 2. The first line shows in bold the title of the bookmark which has the URL of the bookmark as underlying hyperlink. The second line shows an optional description the user can assign to every post. The last two lines belong together and show detailed information: first, all the tags the user has assigned to this post (*web, service,*

[10] `http://www.bibsonomy.org`

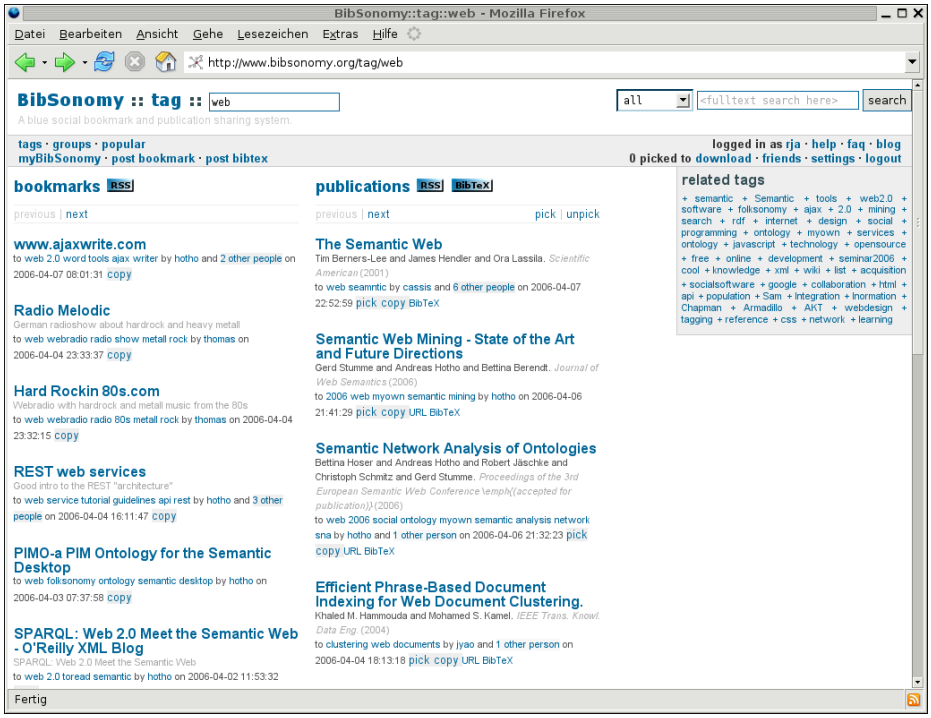

Fig. 1. BibSonomy displays bookmarks and BibTeX based bibliographic references simultaneously.

tutorial, *guidelines* and *api*), second, the user name of that user (*hotho*) followed by a note, how many users tagged that specific resource. These parts have underlying hyperlinks, leading to the corresponding tag pages of the user (`/user/hotho/web`, `/user/hotho/service`, ...), the users page (`/user/hotho`) and a page showing all four posts (i. e., the one of user *hotho* and those of the 3 other people) of this resource (`/url/`r). Section 4.2 will explain the paths given in brackets further. The last part shows the posting date and time followed by links for actions the user can do with this post – depending on if this is his own (*edit*, *delete*) or another user's post (*copy*).

The structure of a publication post displayed in BibSonomy is very similar, as seen in Figure 3. The first line shows again the title of the post, which equals the title of the publication in BibTeX. It has an underlying link leading to a page which shows detailed information on that post. This line is followed by the authors or editors of the publication, as well as journal or book title and the year. The next lines show the tags assigned to this post by the user, whose user name comes next followed by a note how many people tagged this publication. As described for bookmark posts, these parts link to the respective pages. After the date and time the user posted this entry follow the actions the user can do, which in this case include picking the entry for later download, copying it, accessing the URL of the entry or viewing the BibTeX source code.

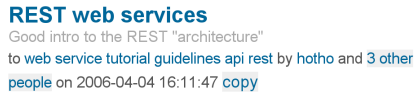

Fig. 2. detail showing a single bookmark post

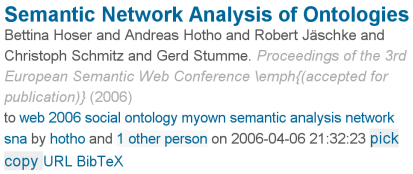

Fig. 3. detail showing a single publication post

4.2 Semantics of the BibSonomy URL scheme

Since group visibility rights (see Section 4.4) make the explanation much more complicated, they are mostly disregarded in this section, as well as in the formal model.

All URLs described here are relative to `http://www.bibsonomy.org`. There are system pages like `/help`, `/settings` or `/post_bookmark` which are necessary for the usage of BibSonomy, but their semantic is straightforward hence they are omitted here. The following list describes the contents C of all pages which show posts in BibSonomy:

`/tag/`$t_1 \sqcup \ldots \sqcup t_n$ Shows every post which has all of the tags $t_1, \ldots, t_n$ attached:

$$C_{t_1,\ldots,t_n} := \{(u, S, r) \in P \mid \{t_1, \ldots, t_n\} \subseteq S\} \tag{1}$$

`/user/`u Shows all posts of user u:

$$C_u := \{(\hat{u}, S, r) \in P \mid \hat{u} = u\} \tag{2}$$

`/user/`u**`/`**$t_1 \sqcup \ldots \sqcup t_n$ Shows every post of user u which has all of the tags $t_1, \ldots, t_n$ attached:

$$C_{u,t_1,\ldots,t_n} := \{(\hat{u}, S, r) \in P \mid \hat{u} = u, \{t_1, \ldots, t_n\} \subseteq S\} \tag{3}$$

`/concept/user/`u**`/`**$t_1 \sqcup \ldots \sqcup t_n$ Shows every post of user u which has for every tag $t \in \{t_1, \ldots, t_n\}$ at least one of its subtags or t itself attached (see also Section 4.4):

$$C_{u,t_1,\ldots,t_n} := \{(\hat{u}, S, r) \in P \mid \hat{u} = u, \forall t_i (i = 1, \ldots, n) \exists t \in S : (\hat{u}, t, t_i) \in \prec \ \vee \ t = t_i\} \tag{4}$$

`/url/`r If r is a bookmark: Shows all posts of the resource r:

$$C_r := \{(u, S, \hat{r}) \in P \mid \hat{r} = r\} \tag{5}$$

`/url/`r**`/`**u If r is a bookmark: Shows the post of user u of the resource r:

$$C_{r,u} := \{(\hat{u}, S, \hat{r}) \in P \mid \hat{r} = r, \hat{u} = u\} \tag{6}$$

`/bibtex/`r If r is a literature reference: Shows all posts of the resource r:

$$C_r := \{(u, S, \hat{r}) \in P \mid \hat{r} = r\} \tag{7}$$

/bibtex/r**/**u If r is a literature reference: Shows the post of user u of the resource r:

$$C_{r,u} := \{(\hat{u}, S, \hat{r}) \in P \mid \hat{r} = r, \hat{u} = u\} \tag{8}$$

/group Shows all groups of the system. More on groups can be found in Section 4.4.

/group/g Shows all posts of all users belonging to the group g:

$$C_g := \{(u, S, r) \in P \mid u \in g\} \tag{9}$$

/group/g**/**t_1␣**...**␣t_n Shows every post which has all of the tags $t_1, \ldots, t_n$ attached and where the user belongs to group g:

$$C_{g,t_1,\ldots,t_n} := \{(u, S, r) \in P \mid u \in g, \{t_1, \ldots, t_n\} \subseteq S\} \tag{10}$$

/viewable/g Shows all posts which are set viewable for members of the group g.

/viewable/g**/**t_1␣**...**␣t_n Shows all posts which are set viewable for members of the group g and which have all of the tags $t_1, \ldots, t_n$ attached.

/search/s Shows all resources, whose full text matches the search expression s, which is interpreted by the MySQL full text search capability.[11]

/download Shows all publication posts which the user has picked in the shopping basket, as described in section 4.4.

/popular Shows the 100 resources posted most often amongst the last 1000 posts. (Note that these numbers are subject to change.)

/ This is the home page of BibSonomy, it shows the entries posted most recently.

An interesting feature, described in section 4.4, is that all paths of URLs described above, can be prepended by a string which changes the output format. In general, posts are shown as HTML lists surrounded by navigation elements and a tag cloud (as seen in Figure 1), but this feature allows the user to get her output in formats like BibTeX or as an RSS feed.

4.3 Architecture

The basic building blocks of BibSonomy are an Apache Tomcat[12] servlet container using Java Server Pages[13] and Java Servlet[14] technology and a MySQL[15] database as backend.

Currently the project has several thousand lines of code and is using the Model View Controller (MVC) [10] programming paradigm to separate the logical handling of data from the presentation of the data. This enables us to produce output in various formats (see Section 4.4), since adding a new output format is accomplished by implementing a JSP as a view of the model.

The database schema of BibSonomy is based on four tables: one for bookmark posts, one for publication posts, one for tag assignments (*tas*) and one for *relations*. Two further tables store information regarding *users* and *groups*. In Figure 4, the two

[11] `http://dev.mysql.com/doc/refman/5.0/en/fulltext-boolean.html`

[12] `http://tomcat.apache.org` [13] `http://java.sun.com/products/jsp`

[14] `http://java.sun.com/products/servlets` [15] `http://www.mysql.com`

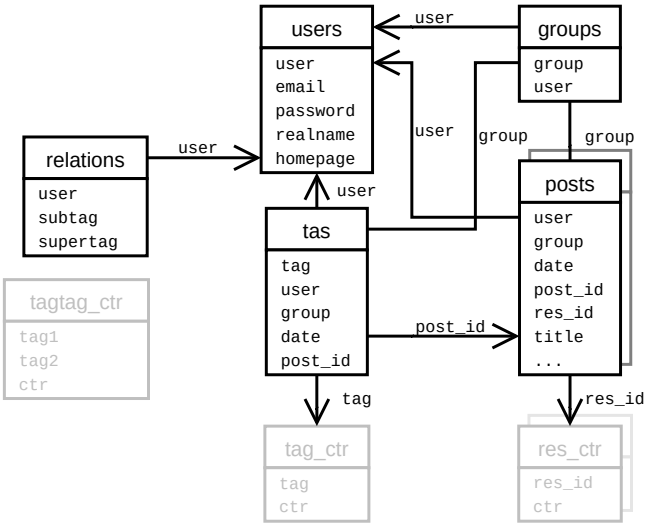

Fig. 4. Relational schema of the most important tables.

posts tables are shown as one and it is only hinted that these are really two tables. The reason to show them as one table *posts* is that they're very similar – the publication post table has just some additional columns to hold all the BIBTEX fields. They are separated in the database for efficiency reasons, since these extra columns just need to be stored for publications.

The posts table is connected with the tas table by the key *post_id*. The scheme is not normalized, on the contrary we have added a high amount of redundancy to speed up queries. For example, besides storing group, user name and date in the posts table, we also store it in the tas table to minimize the rows touched when selecting rows for the various views. Furthermore several other tables hold counters (i. e., how many people share one resource, how often a tag is used, . . .). Finally a lot of indexes (12 in the tas table alone) build the basis for fast answering of queries.

Overall we spent a lot of time investigating and optimizing SQL queries and table schemes and tested both with folksonomy data of up to 8.000.000 posts. At the moment we need no special caching or physical distribution of the database to get reasonable response times, although the system is scalable, since distribution of queries over synchronised databases is possible with MySQL.

4.4 Features

This section describes some extensions of BibSonomy which are not part of the basic system but turned out to be necessary for the everyday use of BibSonomy.

Relations between tags. Tagging gained so much popularity in the past two years because it is simple and no specific skills are needed for it. Nevertheless the longer

people use systems like BibSonomy, the more often they ask for options to structure their tags. A user specific binary relation $\prec$ between tags as described in our model of a Folksonomy (see Section 3) is an easy way to arrange tags. Therefore we included this possibility in BibSonomy.

To enable the addition of elements to the relation already during tagging, we decided to reserve the character sequences `<-` and `->`. That means, if the user u enters t_1`->`t_2, we attach the tags t_1 and t_2 to the respective resource and add the triple (u, t_1, t_2) to the relation $\prec$. The tag t_2`<-`t_1 is interpreted as t_1`->`t_2. Consequently it is not possible to have tags which contain the strings "`<-`" or "`->`". The semantics of this relation is as described in section 3 and can be read as "t_1 *is a* t_2" or "t_1 is a *subtag* of the *supertag* t_2". There are also other ways to add elements to $\prec$, in particular a relation editor.

Usage of this relation is made in several situations. First, the user can structure his tag cloud by showing all subtags of a certain supertag and therefore can see the tags in a hierarchy. Second, BibSonomy offers the option to show on a users tag page not only posts which contain a certain tag, but also posts which contain one of the subtags of the specific tag. This works also for tag intersections: given $t_1, \ldots, t_n$ and a user $u \in U$, then this page contains the set C of posts which is described in Equation 4 in Section 4.2. Compared to *tag bundles* which are available in del.icio.us, this relation is more general and more powerful.

Bringing this relation into the system raises several questions which are still under discussion:

- How to handle cycles, i.e. $u \in U$ and $t_1, \ldots, t_m \in T$ with $(u, t_i, t_{i+1}) \in \prec$ (for $i = 1, \ldots, m-1$) and $(u, t_m, t_1) \in \prec$?
- How to model equivalence or non-equivalence of tags?
- Should we make use of the transitive closure of the relation? If so: where and how to do it efficiently?
- How to express such queries like "all posts which have the tag *howto* and also one of the subtags of *programming*"? One idea involving the reserved character sequences is "`->programming howto`".

Duplicate detection. In particular for literature references there is the problem of detecting duplicate entries, because there are big variations in how users enter fields such as journal name or authors. On the one hand it is desirable to allow a user to have several entries which differ only slightly. On the other hand one might want to find other users entries which refer to the same paper or book even if they are not completely identical.

To fulfill both goals we implemented two hashes to compare publication entries. One is for comparing the entries of a single user (*intra* user hash) and one for comparing the entries of different users (*inter* user hash). Comparison is accomplished by normalizing and concatenating BIBTEX fields, hashing the result with the MD5 [21] message digest algorithm and comparing the resulting hashes. MD5 hashing is done for efficiency reasons only, since this allows for a fixed length storage in the database. Storing the hashes along with the resources in the posts table enables fast comparison and search of entries.

The intra user hash is relatively strict and takes into account the fields *title*, *author*, *editor*, *year*, *entrytype*, *journal* and *booktitle*. This allows one to have articles with the

same title from the same authors in the same year but in different volumes (e. g., a technical report and the corresponding journal article).

In contrast, the inter user hash is less specific and includes only the *title*, *year* and *author* or *editor* (depending on what the user has entered).

In both hashes all fields which are taken into account are normalized, i. e., certain special characters are removed, whitespace and author/editor names normalized. The latter is done by concatenating the first letter of the first name by a dot with the last name, both in lower case. Persons are then sorted alphabetically by this string and concatenated by a colon.

The current duplicate detection is very simple and fails to detect spelling errors, differences in how special characters (like German umlauts) are entered or additional LaTeX commands. This is ongoing work; our implementation allows for simple addition of new hashes.

Currently, duplicate detection is used on the one hand to warn the user when she wants to add an already existing resource and on the other hand to show how many users tagged a certain resource. A step beyond detecting duplicates could be providing the user with additional fields found in other entries referring to the same publication so that she can complete her own entries with additional information.

For bookmark entries in BibSonomy, their URLs are currently just hashed with MD5 and this hash is used for comparison. As can be seen from discussion with users, opinions on if and how to normalize URLs in such systems differ. On the one hand, URLs like `http://www.w3c.org`, `http://w3c.org` and `http://www.w3c.org/index.html` might denote the same resource, on the other hand they're different URLs and it is not obvious whether they really mean the same resource.

Editing tags. Besides changing the tags of a post by editing it, BibSonomy offers at the moment two other ways of changing the tags of several posts at once.

By preceding the path part of a personal URL (i. e., one where the path starts with `/user` followed by the users own username) with `/beditурl` (or `/beditbib`) one can edit the tags of all bookmarks (or publications) on the page at once. This function is also available through links on the respective pages.

Furthermore we have an $m:n$ tag editor which allows a user to exchange m tags by n other tags. More precisely: given two sets A and B of tags[16] and a user $u \in U$, then the $m:n$ tag editor sets iteratively for every $r \in R_u$ with $A \subseteq \text{tags}(u, r)$:

$$Y := (Y \setminus (\{u\} \times A \times \{r\})) \cup (\{u\} \times B \times \{r\}) .$$

Both functions support the user in creating and maintaining a consistent tag vocabulary.

Import of resources. To encourage users to transition from other systems we implemented an import functionality. For del.icio.us, this functionality also takes into account the del.icio.us *bundles*. They are mapped to BibSonomy's relation $\prec$ in the following

[16] If B contains tags not already included in T, then T is adjusted in the obvious way.

way: for every bundle B (which is a set of tags) with name b we add $\{b\} \cup B$ to T and set

$$\prec := \prec \cup (\{u\} \times B \times \{b\})$$

where $u \in U$ is the user these bundles belong to. Furthermore it is possible to import bookmark files of the Firefox[17] web browser, where the typical folder hierarchy of the bookmarks can be added to the users $\prec$ relation. That means that, for every folder a and every subfolder b of a in Firefox, we add (u, b, a) to the users u is-a relation $\prec$, if the user chooses to do so.

Import of existing BIBTEX files is also simple: after uploading the file, the user can tag the entries or automatically assign them the tag *imported*. If a BIBTEX entry contains a field *keywords* or *tags*, its contents are attached as tags to the resource and added to the system. BIBTEX-Fields unknown to BibSonomy are saved in the *misc* field and will not get lost.

Export of resources. Exporting BIBTEX is accomplished by preceding the path of an URL with the string `/bib` – this returns all publications shown on the respective page in BIBTEX format. For example the page `http://www.bibsonomy.org/bib/search/text+clustering` returns a BIBTEX file containing all literature references which contain the words "text" and "clustering" in their fulltext.

More general, every page which shows posts (see Section 4.2) can be represented in several different ways by preceding the path part of the URL with the string described here:

`/` the typical HTML-View with navigation elements
`/xml` bookmarks in XML format
`/rss` bookmarks as RSS feed
`/bib` publications in BIBTEX format
`/endnote` publications in EndNote[18] format
`/publ` publications in a format suited for integration into a homepage (for an integration example see `http://www.kde.cs.uni-kassel.de/schmitz/publikationen.html`)
`/publrss` publications as RSS feed

For example, the URL `http://www.bibsonomy.org/publrss/tag/fca` represents an RSS feed showing the last 20 publications tagged with the tag *fca*.

These export options simplify the interaction of BibSonomy with other systems. RSS feeds allow easy integration of resource lists into web sites or RSS aggregators and BIBTEX output can be used to automatically generate publication lists for papers (as done with this paper). With the help of XSLT it is also possible to transform the RSS output into formats suitable for In addition further formats are implemented easily by extending the URL scheme and adding an appropriate JSP which generates the output.

Currently we are investigating an advanced RDF format for describing literature entries, called Biblio,[19] which has some advantages over BIBTEX. More information regarding Biblio can be found in Section 4.5.

[17] `http://www.mozilla.com/firefox/`
[18] `http://www.endnote.com`
[19] `http://xbiblio.sourceforge.net/biblio/`

Groups. In many situations it is desirable to share resources only among certain people. If the resources can be public, then one could agree to tag them with a special tag and use that tag to find the shared resources. The disadvantage is, that this could be undermined by other users (or spammers) by using the same tag. To solve this problem and also to allow resources to be visible only for certain users, we introduced *groups* in BibSonomy which gives users more options to decide with whom they share their resources.

It is thus possible to have private posts, which only the user can see, as well as posts which can be seen only by group members. Overall there are several aspects of groups in BibSonomy:

1. One can get an aggregated view of all resources of the group members. For example, the URL `http://www.bibsonomy.org/group/kde/seminar2006` represents all posts the members of the *kde* group have tagged with *seminar2006*.
2. It is possible to use groups for privacy so that certain references can only be seen by group members.
3. Resources can be copied directly to the group so that they're persistent, even if a user leaves the group. This is possible, since groups are implemented as a special user which has the name of the group and owns the copied references; this user is also the group admin. This feature is in particular useful where the donator has to commit to the resource, e. g., for project deliverables or student projects.

While we are using one group for sharing resources within our institute, we run several other groups which are used for teaching – they collect resources for students which they might find useful for their lecture.

Shopping Basket. Every publication can be "picked" and is then available in a "shopping basket"-like download area. This is useful for collecting references one needs for a publication. Since all publication related export options mentioned in Section 4.4 apply to this, it is straightforward to get all collected posts in BIBTEX or EndNote format.

4.5 Future Enhancements

Algorithmic Approaches. Since the driving force behind the development of BibSonomy was our need for a system to design, test and integrate new algorithms in the upcoming field of social resource sharing systems, we have already developed algorithms for folksonomies [9, 23]. Currently we plan to integrate these into BibSonomy. Additionally we are investigating further methods to improve such systems.

First of all there is a need expressed by several users to get tag recommendations when they tag a resource. This helps to get an emergent vocabulary and a common understanding and usage of tags between all users. Currently this is work in progress – we already have a recommender for tags implemented which works pretty well and currently we're integrating it into the system. It is based on collaborative filtering [22].

As common in most folksonomy systems, resources in BibSonomy are sorted by date and listed in descending order, newest resources first. But often a ranking by relevance might be more appropriate, especially if not only tags, but also fulltext search

results or subtag queries (see 4.4) are shown. Since our algorithm FolkRank [9] is well suited for that task it will be integrated into BibSonomy. It remains to study, though, how to compute the FolkRank in real time.

Another interesting topic is the discovery of communities of users. For this task there are several methods suitable. Besides standard clustering techniques, a triadic form [13] of Formal Concept Analysis [6] fits well for the structure of a folksonomy. A first implementation of the triadic Next Closure algorithm [5, 11] showed, that it is possible to compute clusters of users which tagged the same resources with the same tags. On a dataset with $|U| = 3301$, $|T| = 30461$, $|R| = 220366$ and $|Y| = 616819$ we got several thousand concepts. The most interesting ones were cuboids with every edge larger than one. But since this approach calculates only exact cuboids, less strict approaches might gain broader results.

If one might inspect the community of users around a specific tag or resource (or even user), the top-k ranked results of FolkRank are a good point to start. With this general ranking algorithm, the detection of topic/resource/user centric communities is possible. Still, some open questions remain to be researched:

- What exactly constitutes a community in a folksonomy?
- Can different kinds of communities be distinguished?
- Which elements of a folksonomy should be used with FolkRank to determine a community?

One further aspect to mention is the area of ontology merging and alignment. Since users personomies can be regarded as lightweight ontologies, one could think of either merging several personomies into one large ontology or trying to align the different personomies.

WebDAV access. With WebDAV[20] it will be possible to access BibSonomy like a file system where every tag is a directory and every resource a file. Since file systems are typically structured hierarchically like a tree – in contrast to a folksonomy, which is a graph – it is an interesting task to map the graph to a tree. At the time of this writing, an early version of a WebDAV interface to BibSonomy is already running and we're discussing details of the mapping. The result will be a lightweight tagging filesystem, comparable to approaches proposed in [20, 1, 3].

Biblio. Biblio[21] is a proposal for a new data model for representing literature references. It is based on XML and provides three classes: events, agents, and bibliographic reference types. Therefore authors or journals become separate entities and this allows for a better handling than in BibTeX. The approach chosen by Biblio looks very promising and might be a basis for the future data model of BibSonomy. At least the export into the Biblio format should be easy to integrate.

[20] Web-based Distributed Authoring and Versioning – see `http://www.webdav.org`
[21] `http://xbiblio.sourceforge.net/biblio/`

BIBTEX Styles. There exists a vast amount of different styles[22] for BibTeX and it is tempting to use them for generating nicely formatted HTML output. This work could be done in principle by using a tool like latex2html,[23] but this turned out to be too computationally expensive. Nevertheless we think of providing such a service.

API. Experience has shown, that an Application Programming Interface (API) is crucial for a system to gain success. It is something which has been requested by many people and which allows for easy interaction of BibSonomy with other systems. Hence we are currently investigating several approaches to add an API to BibSonomy. Most systems use lightweight APIs similar to the idea of REST [4] which can be used and accessed also by not so experienced programmers. Nevertheless, with SOAP[24] there exists a standard for web services which should also be taken into account. Since the process of defining an API for BibSonomy has just started, this is still an open task.

Information Extraction for publication references import. At the moment, literature references can be imported only from proper BIBTEX source code. This is a strong restriction, since most literature references in the web are not available in BIBTEX format but rather in the form of human readable publication lists. Hence our efforts to enhance import focus on techniques to allow for the import of such resources. We expect promising results from information extraction techniques [19] such as implemented in MALLET [16].

Interaction with Conceptual Structures Tools The import/export options discussed in section 4.4 refer essentially to interoperability of BibSonomy with tools like web browsers or type setting programs (although XML is easily transformed to fit other purposes). To interact with state of the art knowledge management tools, RDF[25] as exchange format would be desirable. Currently we're discussing an appropriate RDF schema to represent the folksonomy. The Biblio format mentioned earlier seems to be a good candidate for the BIBTEX part. Besides extending import and export functionality, the aforementioned API will allow for easier exchange of semantically enriched information.

As research on folksonomies is just starting, there is no standardized representation of folksonomies. Proposals often concentrate on rather weak formats[26]. One could also imagine to visualize, explore and edit one's own personomy in FCA tools like ConExp or Toscana. This demands capabilities on the client side to interact with BibSonomy and is therefore currently out of scope of our work. The API will allow interested developers to set up such functionalities as "mashups" upon BibSonomy.

---

[22] http://www.cs.stir.ac.uk/~kjt/software/latex/showbst.html

[23] http://www.latex2html.org/ [24] http://www.w3.org/TR/soap/

[25] Resource Description Framework http://www.w3.org/RDF/ [26] cf. the discussion on http://thecommunityengine.com/home/archives/2005/03/xfolk_an_xhtml.html

5 Related Work

There are currently only few scientific publications about folksonomy-based web collaboration systems available. The main discussion on folksonomies and related topics is currently taking place on mailing lists only.[27]

Among the rare exceptions are [8] and [14] who provide good overviews of social bookmarking tools with special emphasis on folksonomies, and [15] who discusses strengths and limitations of folksonomies. Recent papers include [7] and [2] which focus on analyzing and visualizing the structure of folksonomies.

In [17], Mika defines a model of semantic-social networks for extracting lightweight ontologies from del.icio.us. Besides calculating measures like the clustering coefficient, (local) betweenness centrality or the network constraint on the extracted one-mode network, Mika uses co-occurence techniques for clustering the folksonomy.

6 Conclusion and Outlook

In this paper we described a formal model for folksonomies upon that our own system is built. We presented the main features of BibSonomy, as well as the user interface, basic architecture and future enhancements we are working on.

Since a folksonomy is a rich conceptual structure there are several ways to examine it. Up to now we focused mainly on the graph structure of a folksonomy and exploited and enhanced existing algorithms. With the growing amount of users and the availability of relations between tags, more sophisticated algorithms are needed. With the help of BibSonomy we are able to develop and test them, and let the users profit from our results.

When folksonomy-based systems grow larger, user support has to go beyond enhanced retrieval facilities. Therefore, the internal structure has to become better organized. An obvious approach for this are semantic web technologies. The key question remains though how to exploit their benefits without bothering untrained users with their rigidity. We believe that this will become a fruitful research area for the Semantic Web community for the next years.

The combination of bookmarks and publication entries makes BibSonomy especially valuable for researchers, since they have typically a large amount of these resources to organize and share. Hence our existing and planned enhancements of BibSonomy are targeted to ease the task of managing such resources in a professional way. At last because we are researches, too and expect an immediate improvement of the organization of our own work.

Acknowledgement. Part of this research was funded by the EU in the Nepomuk project (FP6-027705).

[27] for example `http://lists.tagschema.com/mailman/listinfo/tagdb`, the TagDB Mailing List or `http://lists.sourceforge.net/lists/listinfo/connotea-discuss`, the Connotea Mailing List

References

1. Stephan Bloehdorn and Max Völkel. TagFS – Tag Semantics for Hierarchical File Systems, 2006. Submitted for publication.
2. M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, and A. Tomkins. Visualizing tags over time. In *Proceedings of the 15th International WWW Conference*, May 2006.
3. S. Ferré and O. Ridoux. Searching for objects and properties with logical concept analysis. In H. S. Delugach and G. Stumme, editors, *Proceedings of the 9th International Conference on Conceptual Structures*, volume 2120 of *Lecture Notes in Artificial Intelligence*, pages 187–201. Springer, 2001.
4. Roy T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.
5. B. Ganter. Algorithmen zur Formalen Begriffsanalyse. In B. Ganter, R. Wille, and K. E. Wolff, editors, *Beiträge zur Begriffsanalyse*, pages 241–254. B.I. Wissenschaftsverlag, 1987.
6. B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, 1999.
7. Scott Golder and Bernardo A. Huberman. The structure of collaborative tagging systems. Technical report, Information Dynamics Lab, HP Labs, Aug 2005.
8. Tony Hammond, Timo Hannay, Ben Lund, and Joanna Scott. Social Bookmarking Tools (I): A General Review. *D-Lib Magazine*, 11(4), April 2005.
9. Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In *Proceedings of the 3rd European Semantic Web Conference*, Lecture Notes in Computer Science. Springer, 2006. (to appear).
10. G. E. Krasner and S. T. Pope. A cookbook for using the model-view controller user interface paradigm in Smalltalk-80. *Journal of Object Oriented Programming*, 1(3):26–49, 1988.
11. S. Krolak-Schwerdt, P. Orlik, and B. Ganter. TRIPAT: a model for analyzing three-mode binary data. In H. H. Bock, W. Lenski, and M. M. Richter, editors, *Studies in Classification, Data Analysis, and Knowledge Organization*, volume 4 of *Information systems and data analysis*, pages 298–307. Springer, Berlin, 1994.
12. Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, 1986.
13. F. Lehmann and R. Wille. A triadic approach to formal concept analysis. In G. Ellis, R. Levinson, W. Rich, and J. F. Sowa, editors, *Conceptual structures: applications, implementation and theory*, volume 954 of *Lecture Notes in Artificial Intelligence*, pages 32–43. Springer Verlag, 1995.
14. Ben Lund, Tony Hammond, Martin Flack, and Timo Hannay. Social Bookmarking Tools (II): A Case Study - Connotea. *D-Lib Magazine*, 11(4), April 2005.
15. Adam Mathes. Folksonomies – Cooperative Classification and Communication Through Shared Metadata, December 2004. `http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html`.
16. Andrew Kachites McCallum. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu, 2002.
17. Peter Mika. Ontologies are us: A unified model of social networks and semantics. In *Proceedings of the 4th International Semantic Web Conference*, volume 3729 of *Lecture Notes in Computer Science*, pages 522–536. Springer, 2005.
18. Oren Patashnik. BibTeXing, 1988. (Included in the BIBTEX distribution).
19. Fuchun Peng and Andrew McCallum. Accurate information extraction from research papers using conditional random fields. In *HLT-NAACL*, pages 329–336, 2004.
20. Hans Reiser. The naming system venture, 2001. `http://www.namesys.com/whitepaper.html`.

21. R. L. Rivest. The MD5 message digest algorithm. RFC 1321, Apr 1992. `ftp://ftp.rfc-editor.org/in-notes/rfc1321.txt`.
22. Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International WWW Conference*, pages 285–295, 2001.
23. Christoph Schmitz, Andreas Hotho, Robert Jäschke, and Gerd Stumme. Mining association rules in folksonomies. In *Proceedings of the IFCS 2006 Conference*, Lecture Notes in Computer Science. Springer, 2006. (to appear).
24. S. Staab, S. Santini, F. Nack, L. Steels, and A. Maedche. Emergent semantics. *Intelligent Systems, IEEE [see also IEEE Expert]*, 17(1):78–86, 2002.
25. L. Steels. The origins of ontologies and communication conventions in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 1(2):169–194, October 1998.
26. Gerd Stumme. A finite state model for on-line analytical processing in triadic contexts. In Bernhard Ganter and Robert Godin, editors, *Proceedings of the 3rd International Conference on Formal Concept Analysis*, volume 3403 of *Lecture Notes in Computer Science*, pages 315–328. Springer, 2005.
27. Rudolf Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In Ivan Rival, editor, *Ordered sets*, pages 445–470, Dordrecht–Boston, 1982. Reidel.