University of Southampton

Faculty of Engineering, Science and Mathematics

School of Electronics and Computer Science

RichTags: A Social Semantic Tagging System

by

Georgios I. Fountopoulos

20 December 2007

A dissertation submitted in partial fulfilment of the degree of

MSc Web Technology

by examination and dissertation

Abstract

Social tagging systems allow users associating arbitrary keywords (or tags, or labels) to resources they want to save for future recall. Such saved items are called posts or bookmarks and usually constitute shared information in social tagging systems (although access control mechanisms might be applied as well). This means that users of a social tagging system can save and share their bookmarks with each other. The term social stresses the fact that much of the usefulness of the system relies on the data the users submit and share with each other.

As a member of this category of tools, RichTags aims to overcome some weaknesses of the conventional social tagging systems (folksonomies) by utilizing Semantic Web technologies. The defining characteristic of the system is that the tags constitute an ontology of meaningful concepts, which is collectively managed by the users of the system. Hence, the approach is called *social semantic tagging*. It overcomes the *polysemy*, the *synonymy*, and the *basic level* variation problems encountered in the conventional systems. As well, it offers higher *precision* and *recall*.

Current realisation of semantic tagging basically concerns an effort to automatically derive semantics out of folksonomies without affecting the mechanism of tagging applied in them. In contrast, RichTags's approach for semantic tagging is a social process relied on the collective intelligence of the users instead of automation methods. The later means that the users collectively expand the tag vocabulary throughout the tagging task, while consistency mechanisms are applied to keep the vocabulary consistent during this expansion.

The basic factor that differentiates RichTags from existing proposals for the enhancement of tags with meaning is that the primary mechanism relies on human collective intelligence and not on automation methods. However, this does not mean that the proposed automation techniques could not be combined with RichTags; contrariwise they could be very useful to speed up the production of the initial set of semantic tags in the vocabulary.

Finally, RichTags is not limited to enriching the tags with meaning as current efforts primarily aim to; instead it utilizes this semantic information to improve the tagging and the exploration tasks of tagging systems.

Acknowledgements

I would like to thank my sponsor, the Greek State Scholarship's Foundation, for giving me the opportunity to study on this course. My project supervisor, dr monica schraefel, offered advice and guidance for the project and I am grateful for that. Finally, I would like to thank Daniel A. Smith for helping me on the selection of a project topic in the beginning of this effort.

Table of Contents

1 Introduction

Social tagging systems allow users associating arbitrary keywords (or tags, or labels) to resources they want to save for future recall. Such saved items are called posts or bookmarks and usually constitute shared data in social tagging systems (although access control mechanisms might be applied as well). This means that users of a social tagging system can save and share their bookmarks with each other. The diversity of possible motivations for such action has been widely discussed and there have been attempts to analyze how these motivations might affect the kind of keywords produced for a given resource [1].

The term social stresses the fact that much of the usefulness of the system relies on the data the users submit and share with each other. This is what web 2.0 [2] is all about, and social tagging systems are being characterized as web 2.0 applications.

As a member of this category of tools, RichTags aims to overcome some weaknesses of the conventional social tagging systems (folksonomies) by utilizing Semantic Web technologies. Hence, RichTags is called a social semantic tagging system. The term semantic is the defining characteristic of the system and in fact what distinguishes it from other existing approaches (such as Delicious [3], Connotea [4], and Flickr [1]). It means that the tags are not simply free-form strings as they appear in current systems; rather they constitute an ontology of meaningful concepts, which is collectively managed by all the users.

2 Historical overview

The idea of bookmarking can be traced back to the emergence of the web and its first widely accepted web browser, Mosaic, which was primarily released in September 1993 [5]. Mosaic had a feature called *Hotlists*, which allowed a hierarchical organization of links in directories, appeared in a menu within the web browser.

The feature becomes known as *Bookmarks* from the Netscape browser, which was released next year as a commercial application from the same development team.

In August 1995, Microsoft enters the browser market with the release of Internet Explorer, which included a similar link manager called *Favorites*.

The emergence of search engines, such as Yahoo! and Google, made it easier to deal with the huge amount of links available on the web. As bookmark lists were growing, it became evident that it was easier simply to search for a site instead of selecting it from a huge list of bookmarks.

The first social bookmarking tools appear with endeavors like the Open Directory Project [6] and the Yahoo! directory [7], which constituted collaborative efforts to create a shared taxonomy of links, as opposed to the personal hierarchy of links supported by earlier bookmarking tools.

Another development was the bookmarklet, which extended the flexibility of the bookmarking tools. Brendan Eich, who developed JavaScript in 1995 at Netscape, introduced the mechanism. Bookmarklet is a piece of JavaScript code that can be stored as a bookmark link and executed when the link is activated.

The emergence of social tagging systems starts with Delicious [3], which was developed in 2003 by Joshua Schacter. Tagging systems enabled attaching arbitrary keywords (or tags) to bookmarks so to make them more manageable, allowing a search of these bookmarks based on the associated keywords. The set of shared bookmarks and associated tags by many users allowed similar search functionality as the one typically offered by search engines (although the indexing mechanism is different in nature). An overview of a number of social tagging systems appeared by 2005 is given in [5].

On 24th of July 2004, Thomas Vander Wal [8] coined the term *folksonomy* (folks + taxonomy) to represent the method of collaboratively creating and managing tags encountered in the aforementioned social tagging systems.

In 2005, Tim O'Reilly [2] coins the term *web 2.0* to encompass all the web applications that facilitate collaboration and sharing between users, including the social tagging systems.

Finally, RichTags is an ongoing attempt to overcome some weaknesses of the existing social tagging systems by turning the set of flat (pure) tags into an ontology of meaningful concepts (social semantic tagging).

3 Related technologies

3.1 Web 2.0

As have been previously mentioned, social bookmarking is part of a more general realization called web 2.0. The term web 2.0, coined by O'Reilly in 2005 [2], represents the new trend of web applications to facilitate collaboration and sharing between users. Examples of such applications are social-networking sites (like http://www.myspace.com/), wikis (like http://wikipedia.org), and folksonomies (like http://del.icio.us/). The term refers to a change in the way that the web is used rather than any technical change. According to O'Reilly [2], *"Web 2.0 is the business revolution in the computer industry caused by the move to the internet as platform, and an attempt to understand the rules for success on that new platform"*.

The above means that web 2.0 is not something new technically; rather it is a business realization of new ways that the web could be exploited as platform. The companies found ways to support collaboration between users and to benefit financially from the user generated data derived from such collaborations. Technologies like AJAX and Web Services, which defined as key web 2.0 technologies by O'Reilly, existed long before the realization of web 2.0 (e.g. DoubleClick Web Service, DHTML, XHTML & CSS). Furthermore, as evident from [9], many web 2.0 applications in fact encompass features that were originally proposed by the hypertext pioneers.

3.2 Semantic Web

Semantic Web is a vision originally expressed by the creator of the web, Tim Berners-Lee, in 1999 [10]. The vision is to transform the human-understandable content of the today's web into a machine-understandable content, so to enable applications like software agents to find, share, and integrate information more easily. As Berners-Lee states in his book:

> *"I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A 'Semantic Web', which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to*

machines. The 'intelligent agents' people have touted for ages will finally materialize."

In a subsequent article in 2001, Berners-Lee et al. [11] describe a representative application and define the key technologies for the Semantic Web. The application concerns a software agent capable of consulting the user's busy schedule, and other agents running on behalf of medical doctors, in order to present to the user the optimum solution for booking an appointment with a doctor. The application clearly demonstrates the benefits of the Semantic Web.

Some of the enabling technologies defined for the Semantic Web are the Resource Description Framework (RDF) [12], the RDF Schema language [13], and the Web Ontology Language (OWL) [14], along with other standards built on top of them (see Figure 1 below).

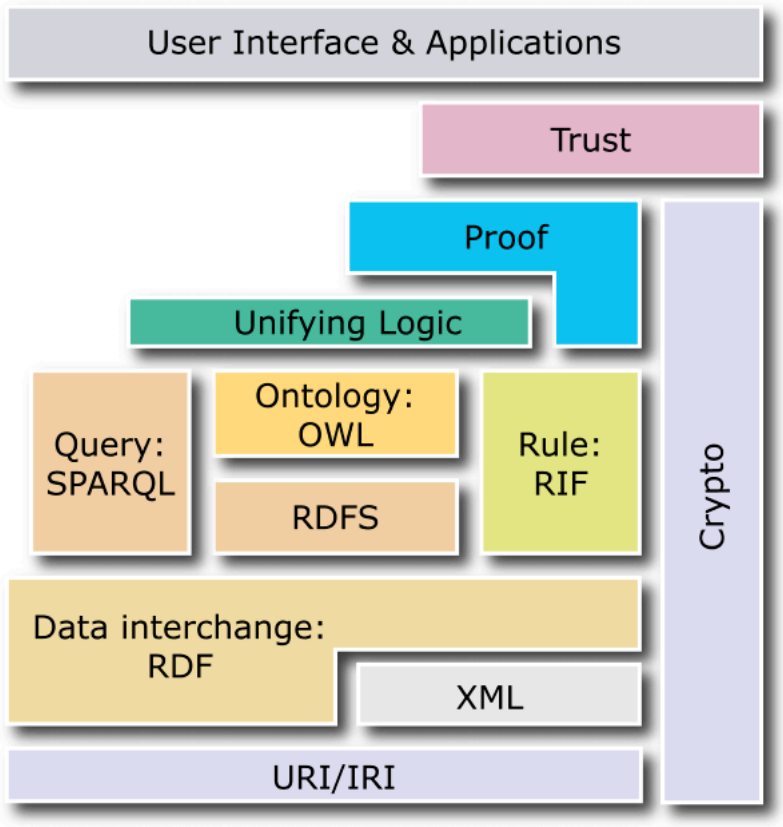

Figure 1. W3C Semantic Web Layer Cake.

The W3C Semantic Web Activity (http://www.w3.org/2001/sw/) is an effort to develop standards and promote the adoption of the Semantic Web.

Recently, the term web 3.0 has been used as synonym to the Semantic Web, and there have been attempts to define subsequent milestones for the evolution of the web with terms like web 4.0 and webos (see Figure 2 below).

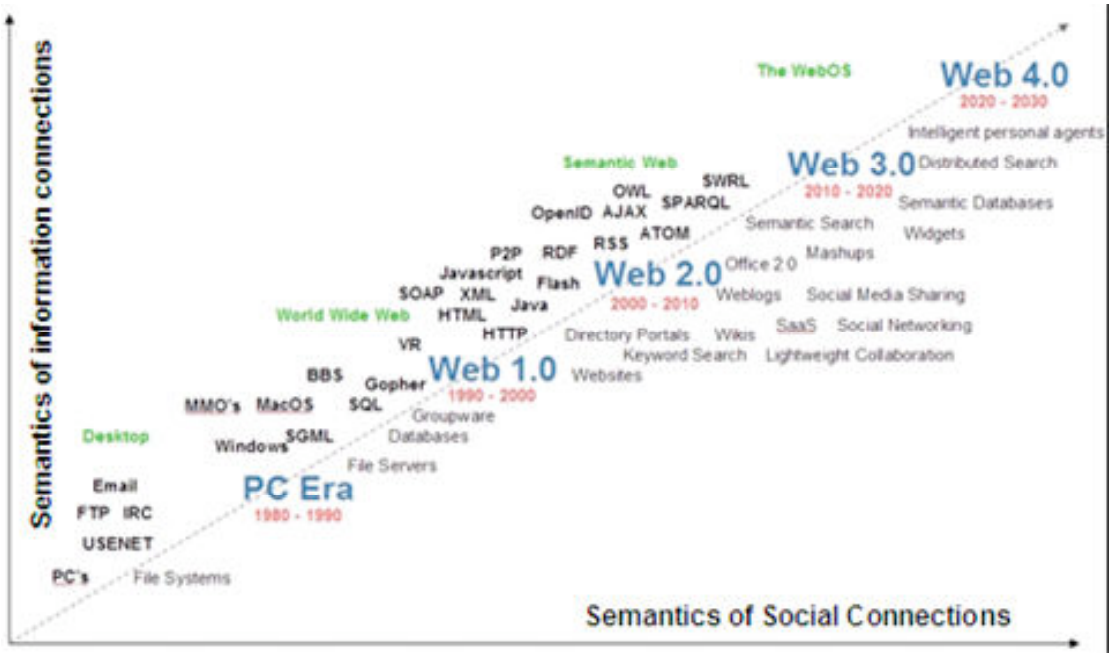

Figure 2. The evolution of the web. From an article on ZDNET (http://blogs.zdnet.com/BTL/?p=4499).

Finally, Grid computing and several other fields can be improved by paying due attention to the Semantic Web [15]. Grid computing is about integrating computing resources; and Semantic Grid is an extension where information and services are given well-defined meaning using Semantic Web technologies [16]. Web Services provide the service-oriented approach for Grid services (OGSA); while Semantic Web Services is an attempt to enable automated discovery, invocation, composition and interoperation, and execution monitoring of the services, using Semantic Web technologies, which allow greater expressivity comparing to WSDL and UDDI [17].

4 RichTags versus current tagging systems

Although many existing tagging systems are targeted for a variety of resource types (such as documents, images, videos, etc), RichTags at the moment is primarily focused on documents, and more specifically on scientific publications, aiming on extending existing tools for academic research. Thus, instead of a somehow general term of resource for the tagged objects, I use the term document throughout this writing.

The functionality of a tagging system can be separated into two basic tasks:

- A user attaches keywords (or tags, or labels) to a document. I call this the *tagging task*.
- A user uses the system to explore the tagged documents. I call this the *exploration task*.

Existing tagging systems (such as Delicious [3], Connotea [4], and Flickr [1]) basically offer the following capabilities for each of the above two main tasks:

- *Tagging task*. When the user tags a document, the system recommends a list of tags based on the tags that other users assigned to the document. The user can select a recommended tag and/or insert a new tag for the document.
- *Exploration task*. The exploration task of current tagging systems offers the following capabilities [18]:
    - Exploration based on a set of tags.
    - Exploration based on the most popular tags in the system.
    - Exploration based on the degree of overlap with a tag the user has entered.

The tags in current tagging systems are flat (pure), meaning they are not connected in any way by some types of relations between them. RichTags improves the two basic tasks by introducing semantic relations between tags. The SKOS ontology [19] is used as a model for expressing such semantic relations between tags. The expressivity of the SKOS vocabulary is indicated in the following subset of SKOS constructs:

- skos:prefLabel (preferred label): The preferred lexical label for a resource, in a given language.
- skos:altLabel (alternative label): An alternative lexical label for a resource.
- skos:broader (has broader): A concept that is more general in meaning.
- skos:narrower (has narrower): A concept that is more specific in meaning.
- skos:related (related to): A concept with which there is an associative semantic relationship.

- skos:definition (definition): A statement or formal explanation of the meaning of a concept.
- skos:scopeNote (scope note): A note that helps to clarify the meaning of a concept.

Using the SKOS ontology as framework, a set of tags and some types of relations between these tags are defined. Such relations include narrower and broader concepts, preferred and alternative labels, scope notes and related concepts (see Figure 3).

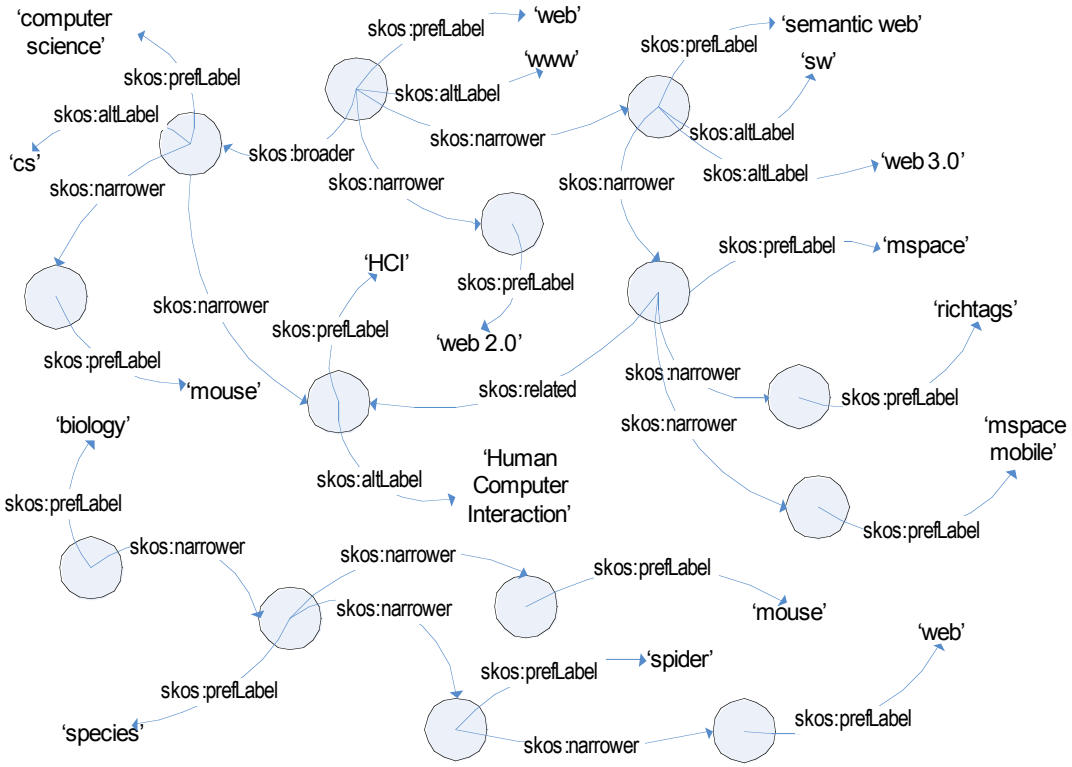

Figure 3. A snippet from a potential tag vocabulary defined using the SKOS ontology.

I call *tag vocabulary* the set of tags enriched with semantic relations between them.

The following Figure 4 presents the approach of the current flat tagging systems.

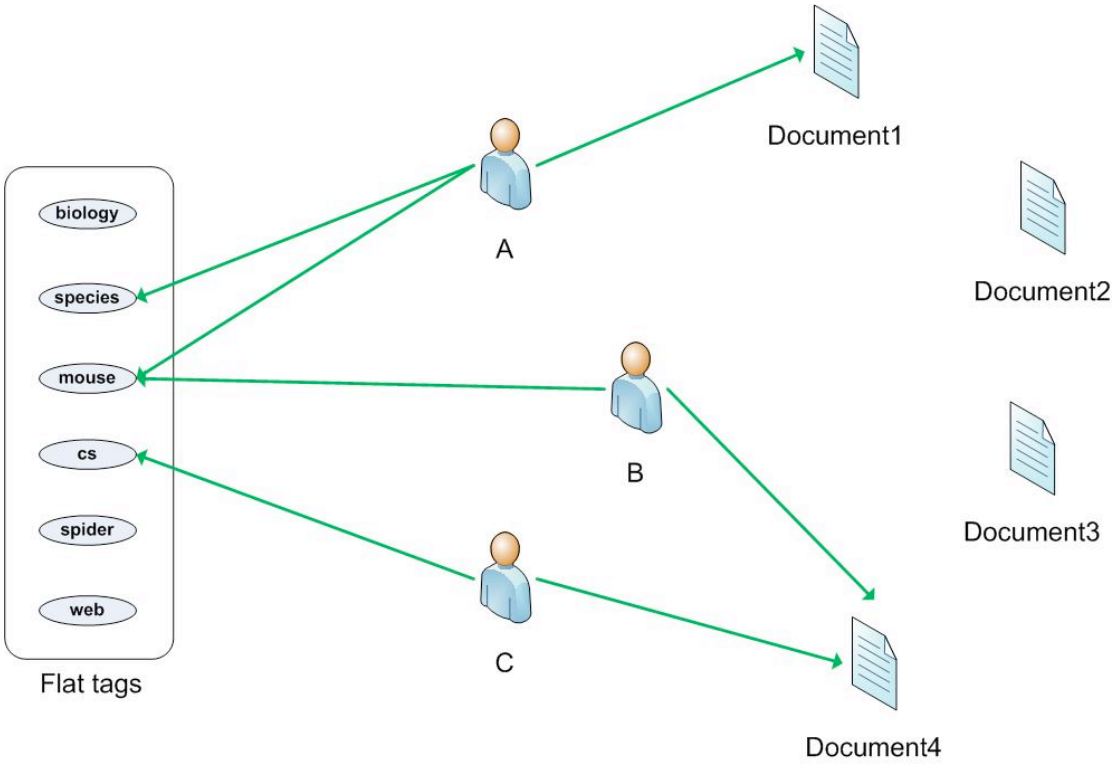

Figure 4. Current flat tagging systems (folksonomies[1]). The tags are not related to each other and do not imply any particular meaning.

Respectively, Figure 5 below shows the approach of our amended tagging system. Instead of having a set of flat tags attached by some users to some documents, a special vocabulary (tag vocabulary) is used in order to enrich the set of tags by adding relations between them and defining their meaning. I call this approach *semantic tagging*.

[1] Note that the term *folksonomy* embodies all the three elements of a tagging system (documents, users, tags) whereas the term *tag vocabulary* refers only to the set of tags in a semantic tagging system.

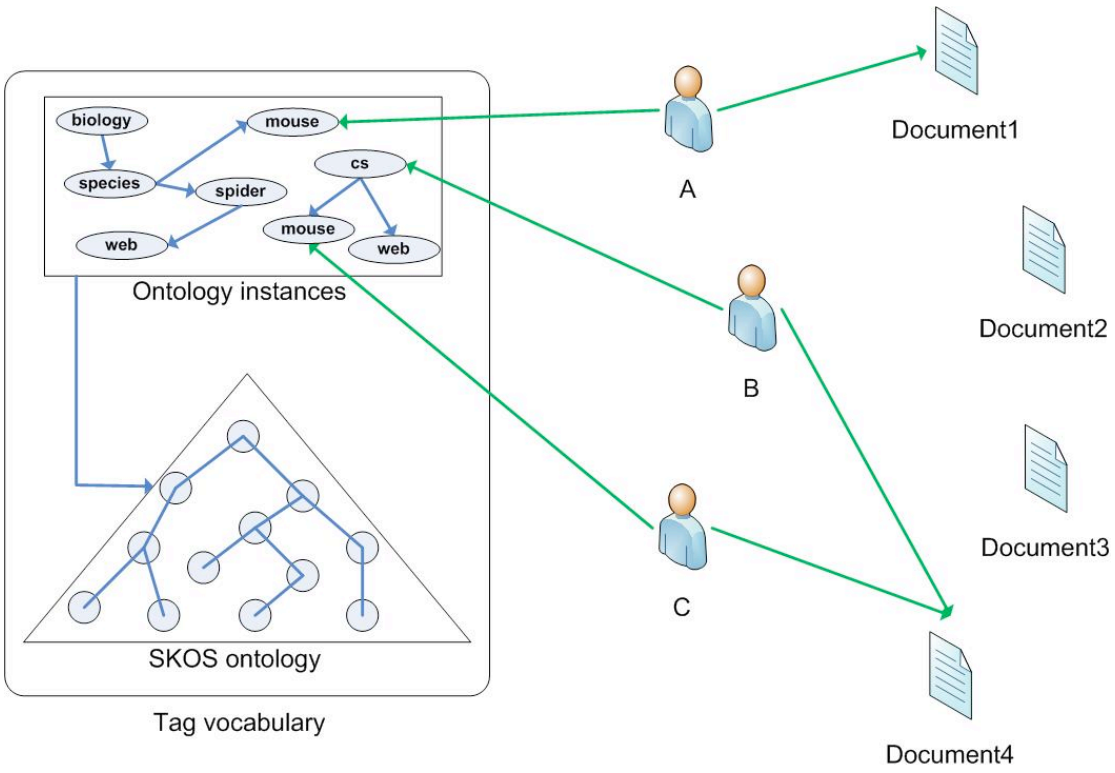

Figure 5. RichTags semantic tagging system. The tag vocabulary specifies relations between tags and attaches meaning to them.

4.1 RichTags in the formal design taxonomy of tagging systems

In 2006, Marlow et al. [1] presented a taxonomy of architectures based on some key design dimensions and user incentives, which a tagging system might support. As they argue, *"different designs and user incentives can have a major influence on the usefulness of information for various purposes and applications, and in a reciprocal fashion, on how users appropriate and utilize these systems"*. To stimulate the understanding of the system, here I will position RichTags in the dimensions of their design taxonomy. I will not extend to the user incentives since RichTags does not restrict to any of those incentives presented in their taxonomy (in fact it supports all of them).

- *Tagging Rights.* According to this dimension, systems are separated to *self-tagging,* where users can tag only the content they create, and *free-for-all*, where there is no such restriction. As well, access control mechanism might be applied to allow varying levels of restriction. RichTags is a *free-for-all* system, thus the users can tag any content no matter who created it. Moreover, it is of particular importance to consider how the tag vocabulary can be

collectively managed by the users, since the tags are not simply free-form strings, but they constitute concepts with semantic relations, which concepts are created and used in common by many participants. To eliminate potential problems derived from this in common management, RichTags forces a number of rules, which currently are as follows: a user cannot delete a concept (or tag) unless he has created it and no one else has used it; a user cannot modify a concept unless he has created it; and finally, if a concept has been used by someone else, then the user cannot modify the preferred label. Lastly, note that the aforementioned rules are applied to the concepts themselves, whereas no one else can modify a user's associations of tags to resources in one's posts (although this might happen automatically when merging concepts, see Section 8.4.2 for a description of the merging action).

- *Tagging Support.* Depending on the mechanism to support the tagging task, systems are separated to *blind tagging*, if a user cannot see the tags other users have entered for the resource; *viewable tagging*, if the user can see the tags associated by others to the resource; and *suggestive tagging*, if the system can recommend the use of some tags for the resource. RichTags is a suggestive tagging system.
- *Aggregation. Bag-model* approach means that the system allows association of duplicate tags from different users for the same resource, whereas *set-model* approach does not allow such repetition. RichTags uses a bag-model approach since everyone's post for a given resource is saved and managed separately.
- *Type of object.* RichTags at the moment is primarily focused on documents. However this does not restrict to any particular resource type, contrariwise other resource types can be tagged as well.
- *Source of material.* RichTags is open for tagging of any resource. That is, there are no restrictions on the source of material to be tagged.
- *Resource connectivity.* The openness for tagging of any resource consequences to no restrictions on resource connectivity. Instead, resources can be interconnected in arbitrary ways.

- *Social connectivity.* RichTags does not currently provide dedicated mechanisms to support social connectivity between users.

5 Related work

The term semantic tagging has been used in a variety of other systems, but what I call here semantic tagging, although closely related to some of the existing approaches, indeed differs considerably. In 2003, Dill et al. [20] developed a system called *SemTag*, which was automatically generating semantic tags out of the content of web pages. This approach is different from RichTags's, where the semantic tags are created by users instead of being automatically generated from the content of documents.

In 2006, Heymann and Carcia-Molina [18] proposed an algorithm for converting a set of flat tags into a navigable hierarchical taxonomy of tags. This approach although trying to enrich the set of flat tags in a way, it does so using automation method based on the existing set of flat tags (folksonomy). Again, this is different from RichTags's approach, where users are the ones specifying any hierarchy in terms of relations and meanings of tags.

Other related approaches try to amend the tags by integrating multiple resources and techniques. In [21] the authors are using online lexical resources, ontologies, and Semantic Web resources in order to enrich the tags with meaning. In [22] the authors combine this technique with deriving actual ontologies out of folksonomies. Although the authors in [22] recommend involving human intelligence in the approval of the automatically obtained semantics of tags, RichTags's approach differs in that it is completely relied on human intelligence for both obtaining and approving of the semantics of tags.

Note that what differentiates RichTags from existing proposals for the enhancement of tags with meaning is that the primary mechanism relies on human collective intelligence and not on automation methods. However, this does not mean that the aforementioned automation techniques could not be combined with RichTags; contrariwise they could be very useful to speed up the production of the initial set of semantic tags in the vocabulary.

Finally, RichTags is not limited to enriching the tags with meaning as the preceding proposals do; instead it utilizes this semantic information to improve the tagging and the exploration tasks of tagging systems.

6 An abstract definition of the problem

RichTags, and in fact any other Information Retrieval system, deals with a problem which can be decomposed into two separate tasks:

- *Discovery of unknown resources (discovery task).* This task takes place when a user wants to find information about something. The user uses various tools in order to accomplish this task. Typical tools include search engines, online directories, and social bookmarking tools. The user is usually presented with a list of results and selects those relevant to his search. Furthermore, the user needs to save the items he selected during this task so to avoid repeating all over again the procedure of selection in a future recall.
- *Recall of known resources (recall task).* Another way a user can use an Information Retrieval system is to recall previously obtained information. Our brain has limited ability of memorizing information. For this reason sometimes we need to recall information we have been previously acquired but do not (precisely) remember anymore. A reasonable Information Retrieval system should make this task easier than the first one, enabling the user to avoid repeating all over again the amount of effort (e.g. filtering) during the discovery task. This is the primary goal bookmarking tools are trying to achieve, by allowing the user to save selected items for future recall during the discovery task.

Both of the above IR tasks can be decomposed further depending on the kind of information the user provides to the IR system in order to get his results. Thus, the retrieval can be either *content-based* or *keyword-based*.

In content-based retrieval the user uses a part of the content of the resource he is looking for in order to get the results. For example, during the discovery task a user might suppose that a particular phrase should be included in the content of the documents he is looking for. Similarly, during the recall task a user might remember

that a particular phrase was included in the content of the document he is trying to retrieve again.

In keyword-based retrieval the user enters a keyword that describes the resource he is looking for. For example, during the discovery task a user might expect that a particular keyword should describe the resource he is looking for. Similarly, during the recall task a user might remember that a particular keyword was assigned to the resource he wants to retrieve again.

Note that a keyword is not always part of the content, and vice versa. For example, we might attach the keyword "sf" to a document about San Francisco, whereas the document itself might not include anywhere the word "sf". Conversely, the content of the document might include the word "history", which might not be used as keyword.

The merit of the social bookmarking tools (and in fact the reason that the term social is tied to them) is that they improve the discovery task by utilizing the information a user enters to support the recall task for himself. The later simply means that a social bookmarking tool allows a user to save the items he selects during the discovery task, and uses this information to support the discovery task for all the users of the system. A user typically attaches some keywords (or labels, or tags, as they might be called) to the resources he wants to save for future recall. A social bookmarking tool uses these keywords to match them against a search query that anyone can submit to the system, thus using users' collective intelligence in the retrieval process.

On the other hand, a typical search engine (such as Google) is primarily used for the discovery task and is mainly relied on content-based retrieval. As well, ranking mechanisms are applied in order to determine the relevance of the resources so to present the most relevant results first [23]. Keyword-based retrieval is of minor importance in today's search engine implementations and is typically supported by the HTML meta tags (although some search engines use keywords from social bookmarking tools as a means to improve their ranking algorithms).

6.1 Content-based versus keyword-based retrieval

The prior definition motivates the expression of some hypotheses.

Content-based retrieval suits well when the collection of resources is particularly large and dynamic (e.g. the web), because the mechanism to support it can be easily

automated (web spiders). On the other hand, keyword-based retrieval requires user's contribution (bookmarks) and is not as dynamic as the content-based approach.

While content-based retrieval offers higher recall[2], keyword-based retrieval rewards with higher precision[3]. Subsequently, the first is more suitable for the discovery task (especially when we want to discover recently published information or when the amount of results is not too big), while the second supports better the recall task (more precise results).

However, during the recall task, we easier associate the content than the keyword with what we want to retrieve; thus, for the recall task, content-based retrieval might be preferred by some users over keyword-based retrieval.

I believe the ideal system would use a mixture of both the content-based and the keyword based technique.

7 A high level architecture

Figure 6 below depicts a high level architecture of the RichTags web application design. The implementation conforms to the Model-View-Controller (MVC) software design pattern [24]. The architecture consists of some client-side libraries (YUI library [25] and RichTags JavaScript library) and some server-side modules, such as the controller servlet, the JSP view, the business logic, and the Jena Semantic Web framework [26]. The Jena framework is used for the interactions with the ontology (part of the *model*). As well, a database stores all the users' preferences and other data used internally by the system (e.g. cached data). All the server-side components of the web application are deployed in a JSP/Servlet container.

[2] Recall is an Information Retrieval term, which means the percentage of retrieved relevant documents within the total amount of the relevant documents. Please do not confuse with the *recall task* I am describing in this document.

[3] Precision is the percentage of relevant documents within the amount of retrieved documents.

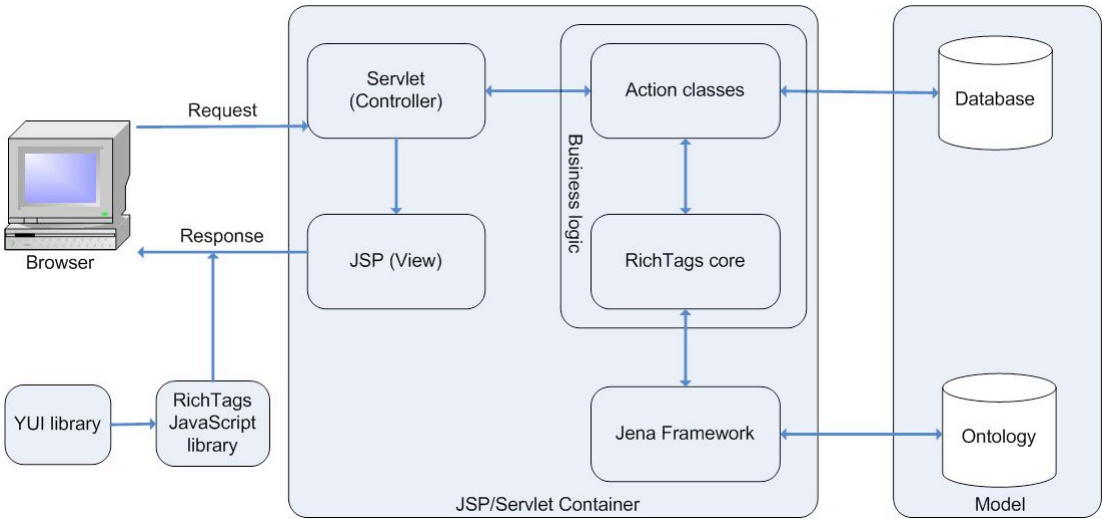

Figure 6. A high level architecture of the RichTags web application design.

The controller servlet handles all the request processing and delegates the requests to corresponding JSP pages for presentation. The business logic consists of the *action classes* and the *RichTags core*. The action classes implement the required logic for serving client requests. The RichTags core implements the main business logic and hides data processing details according to the Data Access Object (DAO) pattern [27]. The later will serve for easier migration to a different data access technology in case such decision will prove being reasonable in future. For example, instead of the use of a simple OWL file and the Jena framework, I am considering a more mature technology for data management, such as a database system that would support exporting to ontology and SPARQL queries. Performance and lack of features are the two main reasons for such consideration. Semantic Web tools are relatively recent and still in research (Berners-Lee et al., 2001 [11]), comparing to the database systems, which have a long history of development and optimization concerning data management (Codd, 1970 [28]). Thus, as a serving example, the Jena SPARQ query language implementation offers a very limited functionality comparing to the SQL query language supported by a typical database today, like MySQL [29]. Some of such missing functionality concerns SQL aggregate functions, nested queries, and referential integrity.

From the data tier perspective, the ontology, which holds the application's data, is available to third party applications in various forms and can be managed using the RichTags Web Service, as shown in Figure 7 below. Thus, all the application's data can be either directly retrieved in raw OWL format, or queried in SPARQL, using the

Joseki SPARQL engine [30], which is integrated into the RichTags web application. In addition to the later two read-only options, the RichTags Web Service enables authenticated third parties to manipulate the ontology. The Web Service is deployed in an Axis2 Web Services engine [31] and offers the following operations for data management:

- *GetTagVocabulary.* Returns the tag vocabulary as a set of concept objects.
- *AddTag.* Creates a new concept in the tag vocabulary.
- *DeleteTag.* Deletes a concept from the tag vocabulary.
- *GetAllPosts.* Returns all the posts made by the user account authenticated for the use of the Web Service.
- *AddPost.* Adds a new post for the authenticated user account.
- *DeletePost.* Deletes a post from the ontology.

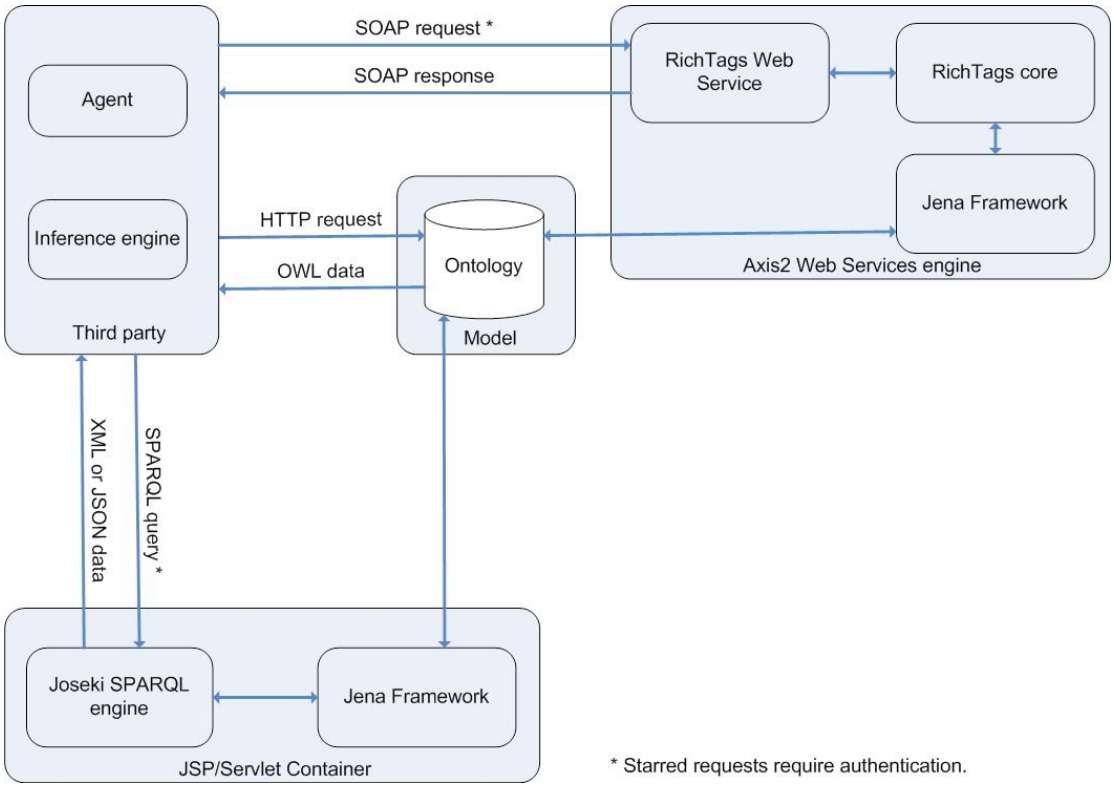

Figure 7. The ontology can be retrieved by third parties in OWL, XML, or JSON format and can be managed using the RichTags Web Service.

Note that the direct access to the OWL data does not require authentication. Hence, any kind of third party application, such as a software agent or an inference engine, which is capable of understanding the OWL syntax, can make unrestricted use of the

application's data (excluding private posts which are stored elsewhere). The later feature makes RichTags a good representative of the kind of applications envisioned to build a Semantic Web [11].

However, the use of the integrated Joseki engine to submit SPARQL queries requires authentication, since it consumes computing resources and otherwise it would make the application susceptible to threats such as denial-of-service (DoS) attacks. Nevertheless, as the ontology is publicly retrievable in raw OWL format, third parties can use other SPARQL engines to query over the OWL data.

Finally, the RichTags Web Service enables not only the access to the ontology, but also the modification of a user account's data by third parties that are authenticated using the particular user's credentials. Moreover, it is the only way for third party applications to access and manage a user's private posts.

8 Contributions

In a formal study of tagging systems in 2005, Golder and Huberman [3] point out some weaknesses of the current implementations (in particular, the Delicious system). Such weaknesses include the polysemy, the synonymy, and the basic level variation problems. The following Sections 8.1, 8.2, and 8.3 describe each of these problems and explain how they have been addressed in the RichTags system, while Section 8.4 outlines some further improvements.

8.1 The polysemy problem

The polysemy problem occurs when a single word has multiple meanings [3]. For example the word "mouse" may mean an input device used with computers, or a small mammal in a biological taxonomy. Similarly, the word "apple" may refer to a fruit, or alternatively to a company's name. Current tagging systems (such as Delicious [3], Connotea [4], and Flickr [1]) cannot express the semantic differences of such polysemous words. This results in lower precision since a query for a polysemous word will return all the items matching to any of the meanings of the word.

Taking the "mouse" as an example of a polysemous word, the search results on Delicious [3] would include items for both the mammal and the input device as shown in Figure 8 below.

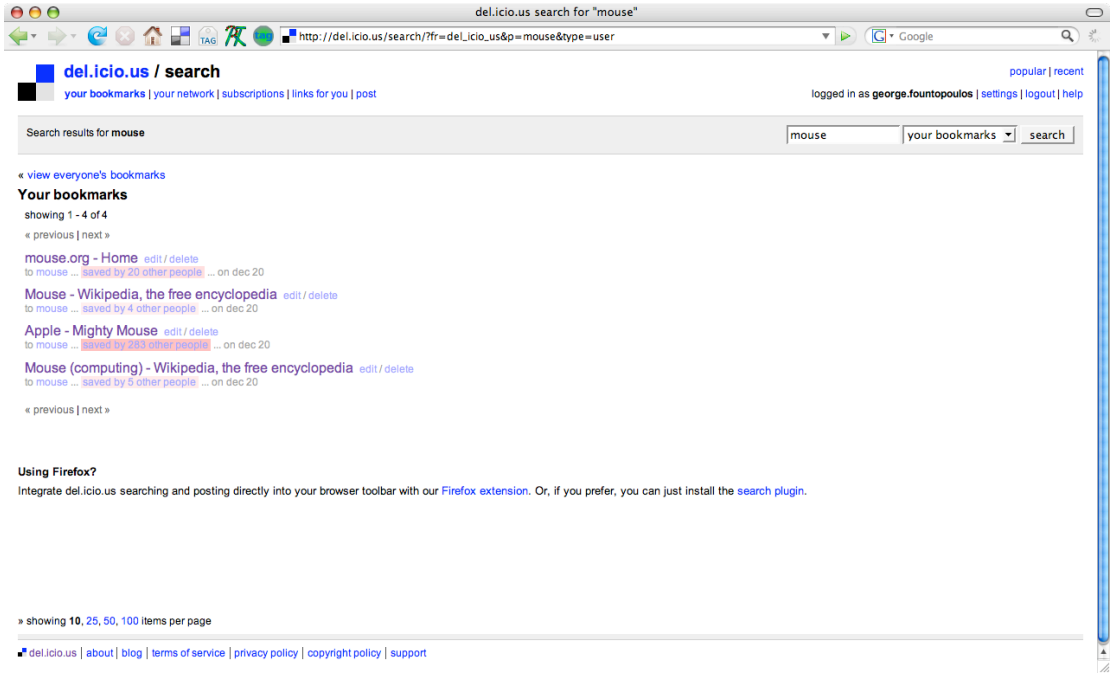

Figure 8. Polysemous words hinder precision in current tagging systems.

In contrast, RichTags would distinguish all the meanings of the word "mouse" and would present them for us to choose. This is demonstrated in Figure 9 below, where, in the section "All Matched Tags", you can see the two distinct concepts that match to the word "mouse".

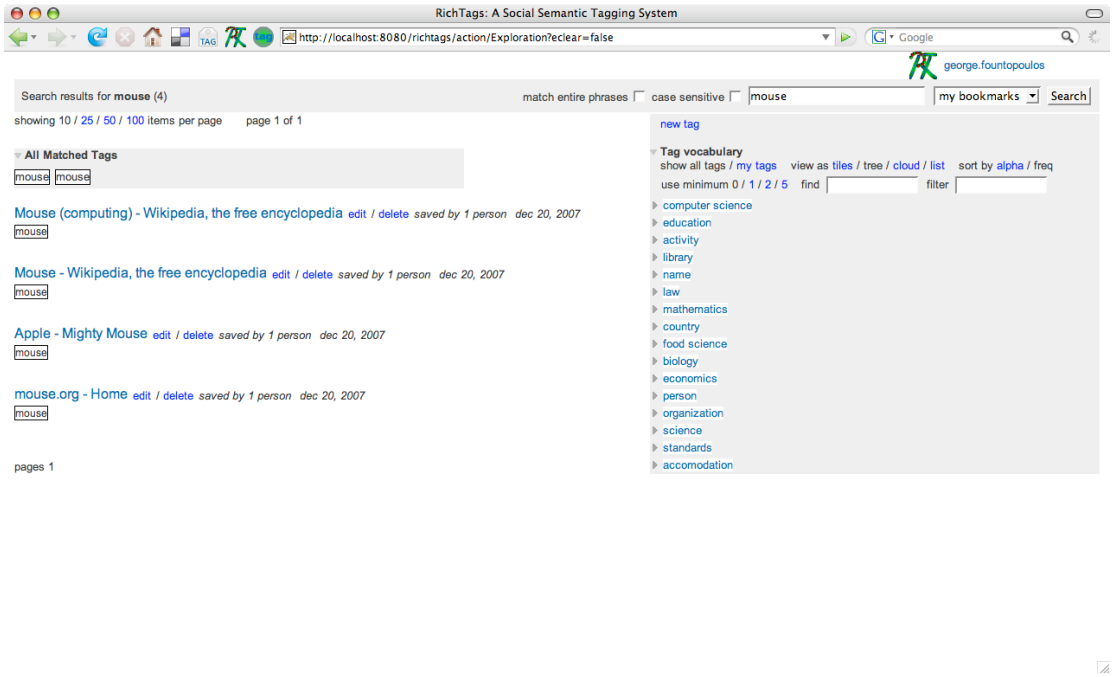

Figure 9. RichTags can distinguish all the meanings of a polysemous word.

By clicking on one of the concepts, some information is shown and a number of options are given for us to choose. As shown in Figure 10 below, the user has clicked on one of the concepts to see that it has one broader, named “hardware”. By clicking on the second concept the user would see that the broader is “species”. This is the way that a user can easily distinguish the exact meaning of a concept (note that the menu shows all the relevant information including all the (directly) broader, narrower, or related concepts and all the alternative labels of the concept).

Figure 10. By clicking on a concept the user can see the associated information (alternative labels and semantic relations) about the concept along with some options.

By clicking on the option “Browse my items” from the menu in Figure 10, the user is getting 100% relevant items to the exact concept he has been chosen (see Figure 11 below). This makes the system achieving 100% precision.

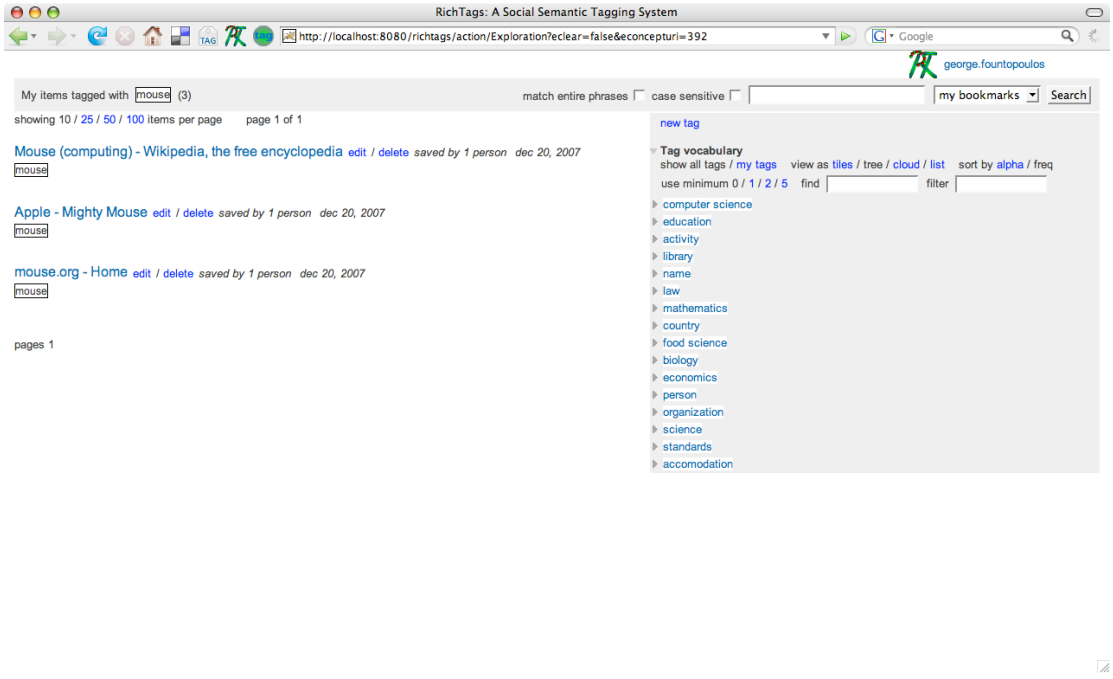

Figure 11. The system achieves 100% precision showing only the items associated with the exact concept the user chose.

Another remarkable feature is that, in addition to the matched tags from a search query, all the tags with narrower meaning are included as well. This does not have any impact on the precision (the precision remains 100%), since documents tagged with a narrower concept are definitely related to the broader concept (although the reverse is not always the case). For example, a search query for "hardware" would include all the documents tagged with the concept "mouse" which is narrower of "hardware" (see Figure 12 below). This can be achieved due to the tag vocabulary, which defines relations between concepts (broader, narrower, related, etc). Current tagging systems do not support it, simply because their tags are free-form strings and do not imply any particular meaning. A search for "hardware" for example, in a conventional tagging system would include only documents tagged with "hardware"; not being able to recognize that documents tagged with "mouse" should be included as well. Thus, the later RichTags's feature improves the recall, since given a search query there are more relevant items returned as results.

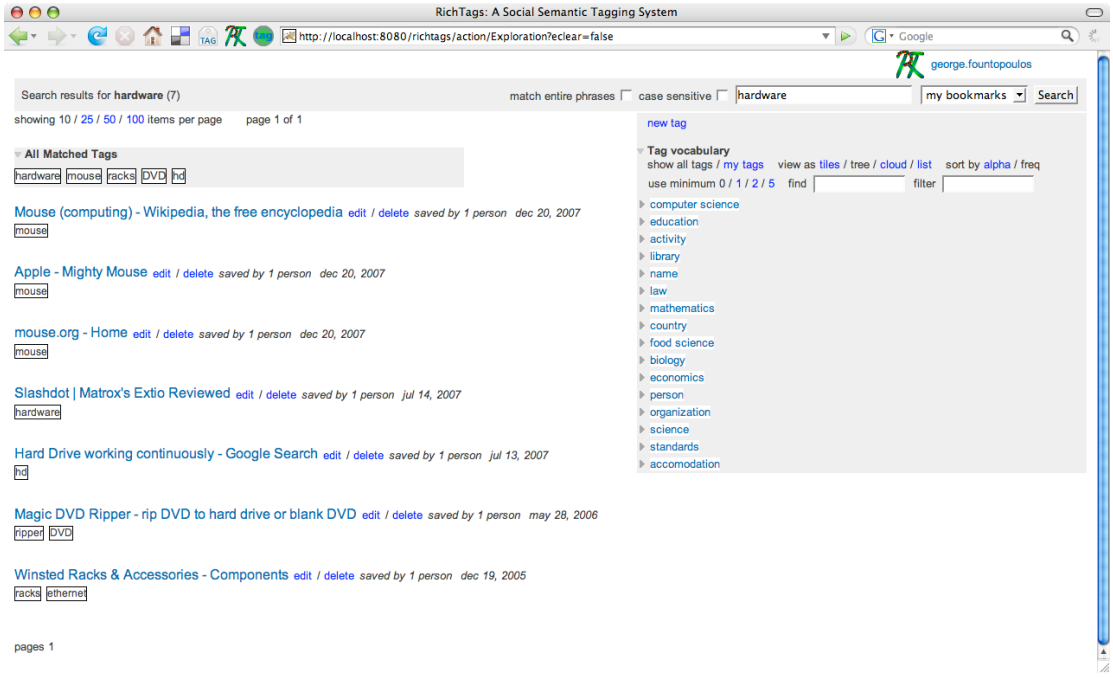

Figure 12. A search query for a concept includes items tagged with any of the narrower concepts. Here a search query for "hardware" includes items tagged with "mouse", which is a narrower concept of "hardware".

Finally, another relevant feature is that a search query can include tags with spaces as shown in Figure 13 below.

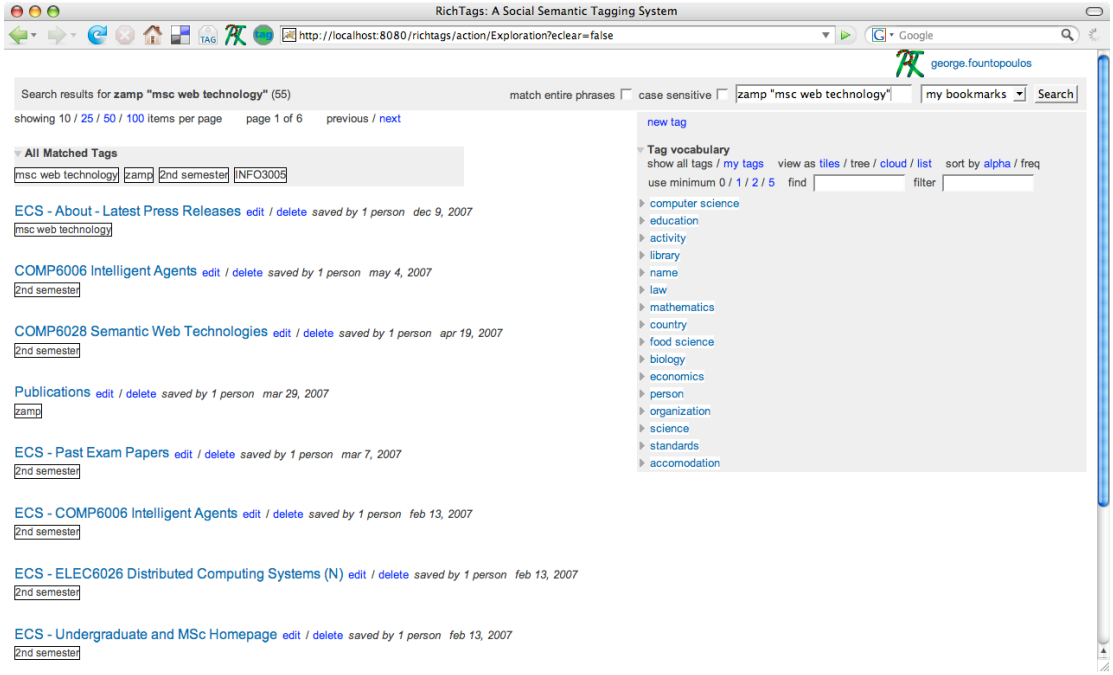

Figure 13. A search query can include tags with spaces. Here the query includes one tag with spaces (msc web technology) and one without (zamp).

8.2 The synonymy problem

The synonymy problem occurs when different words have the same or closely related meaning [3]. For example, the tags “semantic-web”, “sw”, and “web-3.0” may all refer to the same meaning. Plurals and parts of speech and spelling might also constitute a similar problem. One user might use “cat” to tag a document, whereas others might prefer “cats” to tag the same document. Current tagging systems cannot express synonymy of words. Thus, when a user submits “sw” in a query, it is possible that there are items in the system tagged with “semantic-web” or “web-3.0”, which will not be retrieved. A user does not know all the possible variations that other users might have been used for a particular meaning, and even if he does, the system requires to submit all of these variations in order to get all the relevant items. Lower recall is the direct consequence of this problem.

RichTags addresses the problem thanks to the expressivity of the tag vocabulary, which supports multiple labels for a single concept (see Figure 14 below). In particular, the SKOS property skos:prefLabel is used to specify the preferred label and the skos:altLabel is used to specify any number of alternative labels for a single concept.

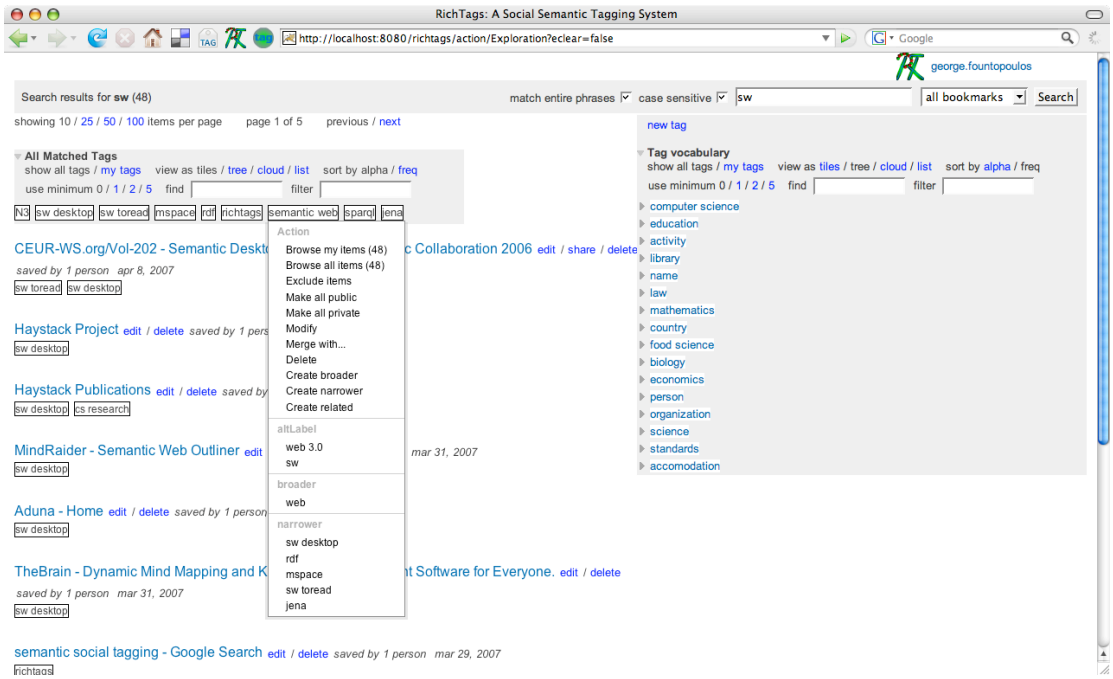

Figure 14. RichTags addresses the synonymy problem thanks to the ability of attaching multiple labels to a single concept. Here you can see a concept with preferred label “semantic web” and alternative labels “web 3.0” and “sw”.

As shown in Figure 14 above, the concept “semantic web” has alternative labels “sw” and “web 3.0”. This enables the system matching the query “sw” to the concept “semantic web”, which has the label “sw” as an alternative label. In fact the system will match to all the concepts that have at least one of their labels matching to the search query. This is the way the synonymy problem is addressed and a higher recall is achieved.

8.3 The basic level variation problem

Different users may use various levels of abstraction to tag a document. A document can be tagged using “cat”, or a more general concept “animal”, or at various more specific levels using “lion” or “tiger”. Current tagging systems do not encourage users using as specific concepts as possible for the tagged items. Furthermore, as discussed in Section 8.1, they cannot recognize that a search query for a general concept like “cat” should include all the items tagged with any of the narrower concepts, such as “lion” or “tiger”.

In contrast, RichTags encourages and makes it easy for the user to select as specific concepts as possible for the tagged items (see Figure 15 below).

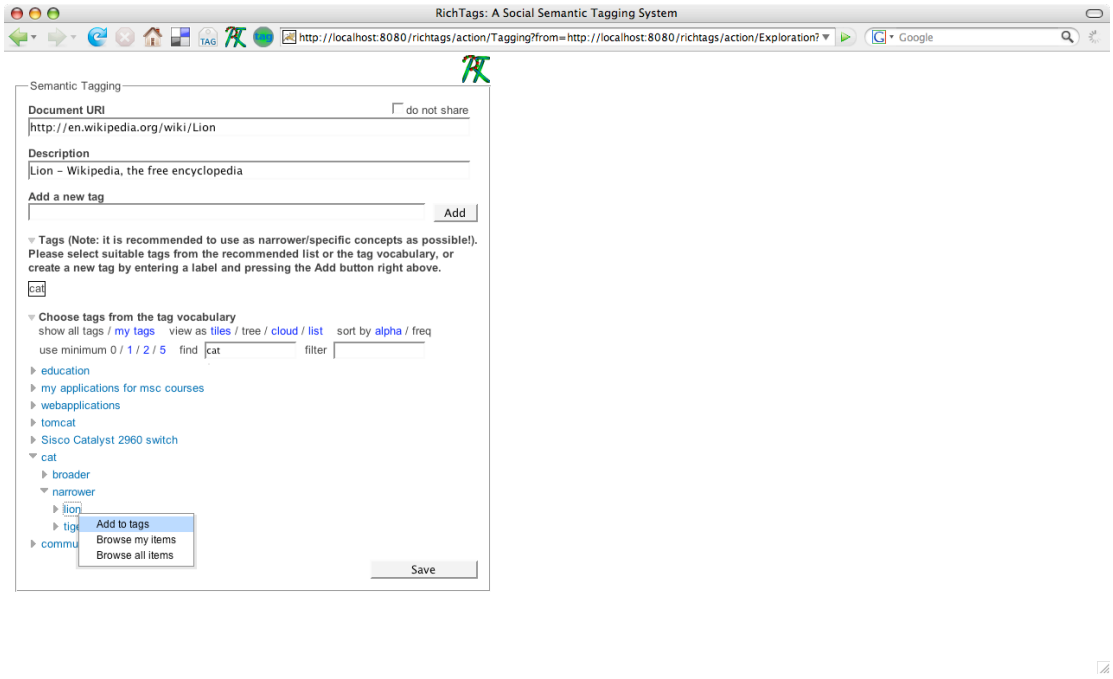

Figure 15. RichTags encourages and makes it easy to select as specific concepts as possible. Simply by clicking on a concept in the tree view the user can see all its narrower concepts.

As can be observed from the above Figure 15, the user can easily find the concept "cat" in the tag vocabulary, and can see all its narrower concepts simply by clicking on it. If the user selects the narrower concept "lion", the system will respond with the message depicted in Figure 16.

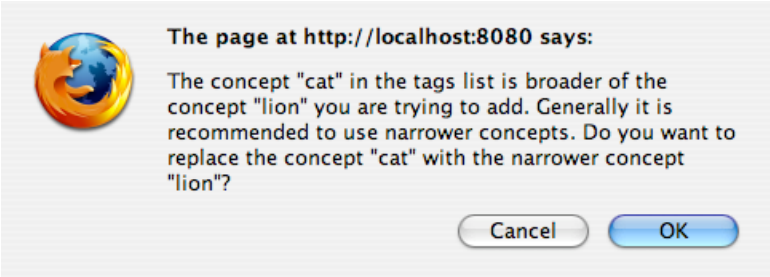

Figure 16. The system encourages and allows automatically replacing a broader concept with a narrower.

Responding positively to the above message will result in Figure 17 below, where you can see that the narrower concept "lion" replaced the broader concept "cat". Thus, the system encourages and helps the user to use as specific concepts as possible for the tagged items. A later query for the broader concept "cat" would include all the items tagged with any narrower concept such as "lion" or "tiger". Respectively, a query for "lion" would return only those specific items tagged with "lion" (or any narrower concepts of "lion" if existent).

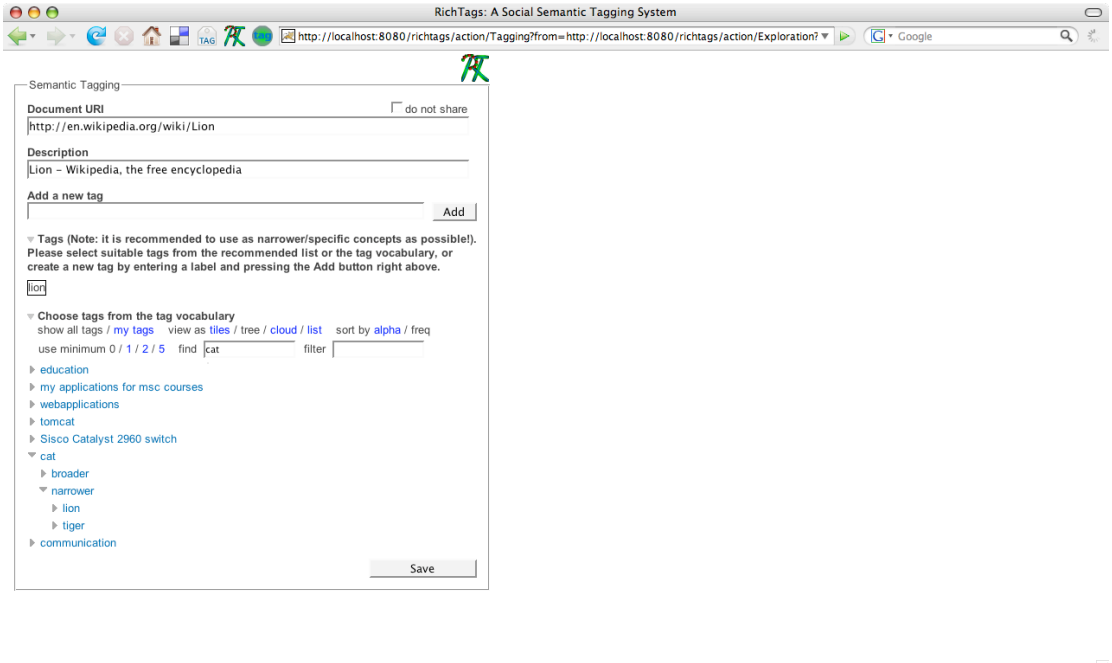

Figure 17. The broader concept "cat" has been replaced by the narrower concept "lion".

Encouraging users using as specific concepts as possible for the tagged documents increases the usefulness of the system allowing higher precision for more specific queries. For example, using specific concepts like “lion” or “tiger” instead of a general concept “cat” enables more specific searches for “lion” or “tiger”, achieving higher precision than the one would be achieved by querying for “cat”.

8.4 Other remarkable improvements

Section 4 outlined the main features of current tagging systems corresponding to the two basic tasks of a tagging system. This Section presents what has been achieved in addition to those features, avoiding the discussion about things already mentioned in previous Sections. Those features discussed in the Sections describing the polysemy, the synonymy, and the basic level variation problems although improve both basic tasks, thought, they are not discussed again here.

8.4.1 Tagging task

RichTags enables the user unambiguously specifying the meaning of the tags he is using when tagging a document. The user can easily determine polysemy, synonymy and levels of abstraction of tags as indicated in Figure 18 and Figure 19 below.

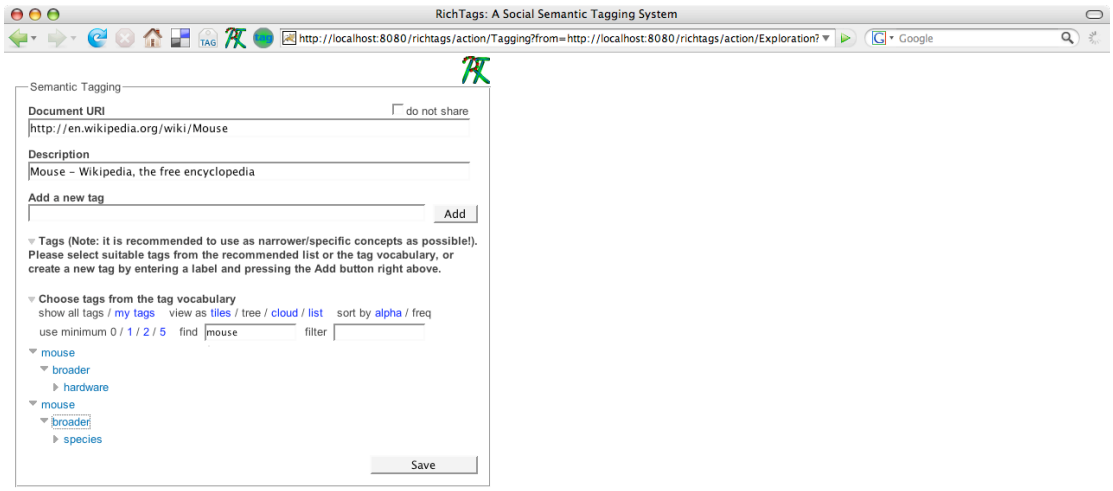

Figure 18. The user can distinguish all the meanings of a polysemous word by looking at the semantic relations. Here the polysemous word “mouse” has two distinct meanings. The one meaning has the concept “hardware” as broader and the other has “species”. Thus, the user can distinguish that the one refers to a device and the other to an animal.

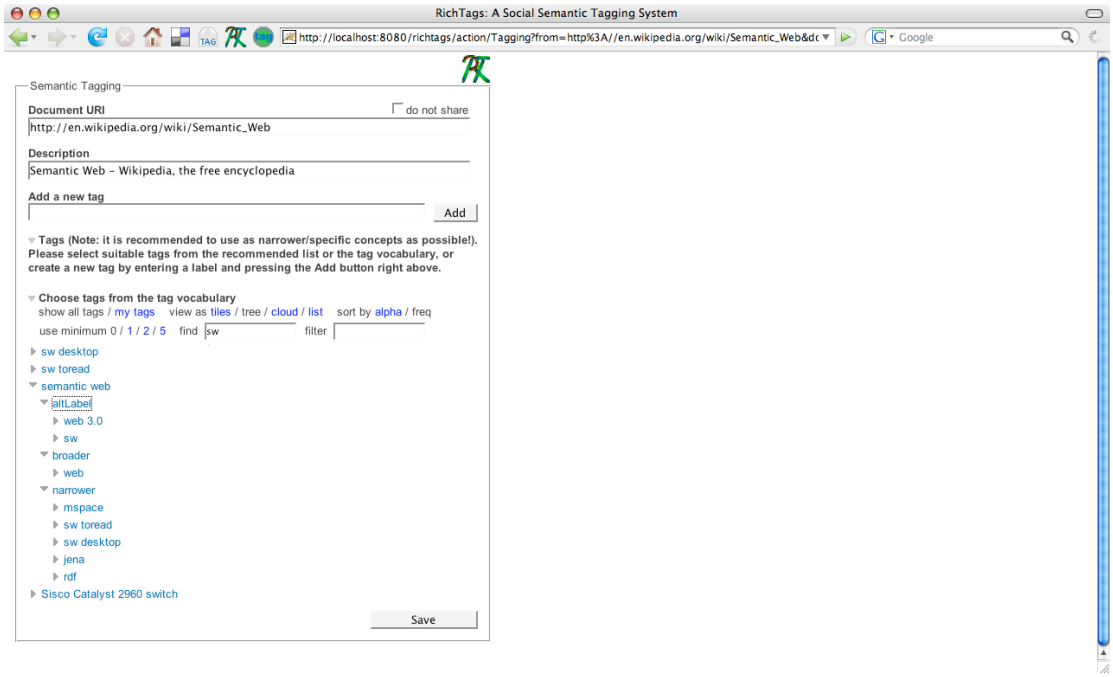

Figure 19. The user can determine the synonymy and levels of abstraction of tags consulting the alternative labels and the semantic relations respectively. Here the concept "semantic web" has alternative labels "sw" and "web 3.0", one broader concept "web", and some narrower concepts like "mspace" and "rdf".

Another improvement concerning the tagging task is that, when the user adds a new tag, RichTags looks to find if the tag matches to any label of the existing concepts in the tag vocabulary. If the tag matches to some of the existing concepts then the system allows the user selecting one of them or alternatively creating a new concept as demonstrated in Figure 20 and Figure 21 below.

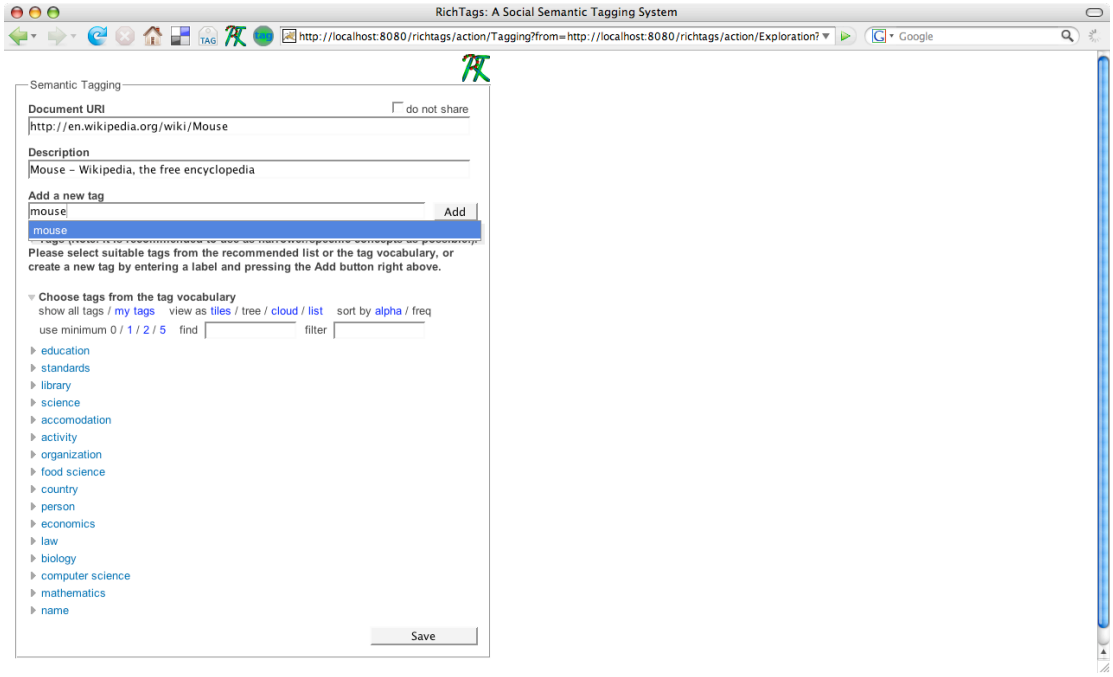

Figure 20. The user adds the tag "mouse" which already exists in the tag vocabulary.

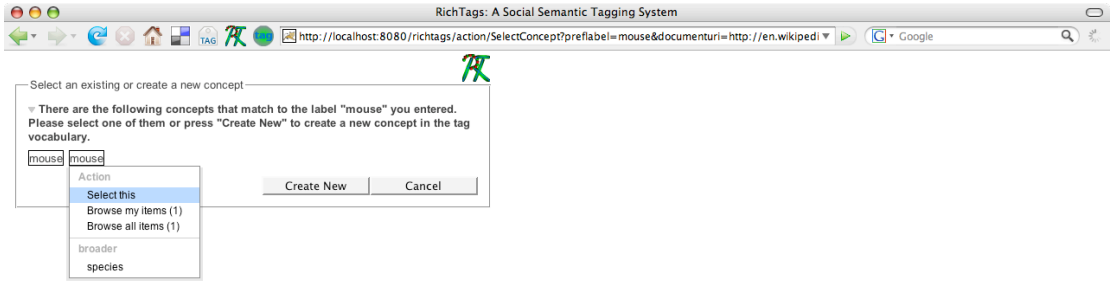

Figure 21. The system finds that there are concepts matching to the tag and allows choosing one of them or creating a new concept in the tag vocabulary.

While creating a new concept (semantic tag) in the tag vocabulary, consistency mechanism is applied so to keep the vocabulary consistent throughout its expansion by the users (see Figure 22 below).

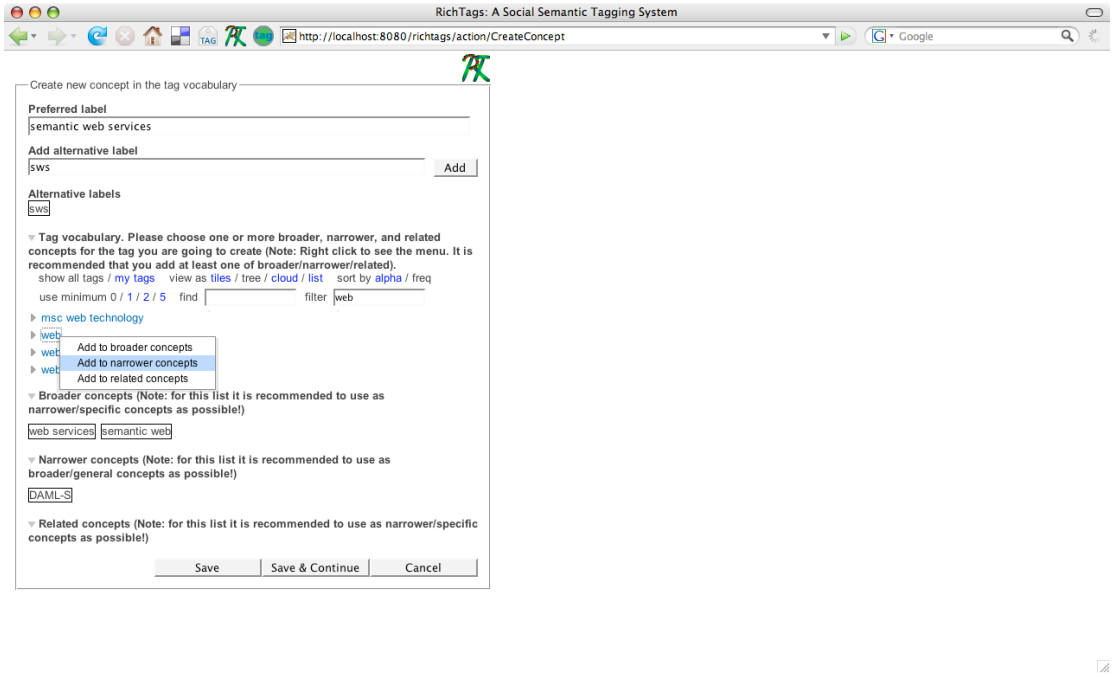

Figure 22. The interface for creating a new concept in the tag vocabulary. The user specifies one preferred label and some alternative labels for the concept. As well, the user can specify broader, narrower, or related concepts from the tag vocabulary. A mechanism is applied to prevent the user defining inconsistent relations.

The user specifies semantic relations for the concept to be created by selecting one or more broader, narrower, or related concepts from the tag vocabulary. The system checks for consistency every time the user specifies a semantic relation for the concept. The rules to keep the tag vocabulary consistent are:

- A single concept can be used only once in a semantic relation. For example we cannot define that a broader concept is narrower as well, or that a related concept is broader as well.
- No broader concept should be narrower of any narrower concept. With different words, no narrower concept should be broader of any broader concept.

If the user enters a relation that do not comply to the above rules, the system preserves the action showing a relevant message analogous to the one in Figure 23 below.

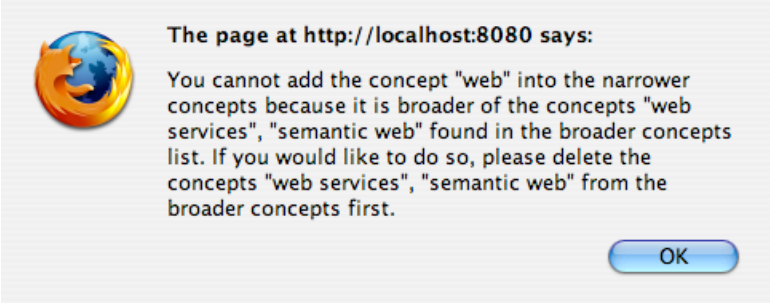

Figure 23. If the user enters an inconsistent relation for the concept the system preserves the action and informs with a relevant message.

Finally, some extra action is required when there are broader concepts that are broader of some concepts from the narrower concepts list, as shown in Figure 24 below.

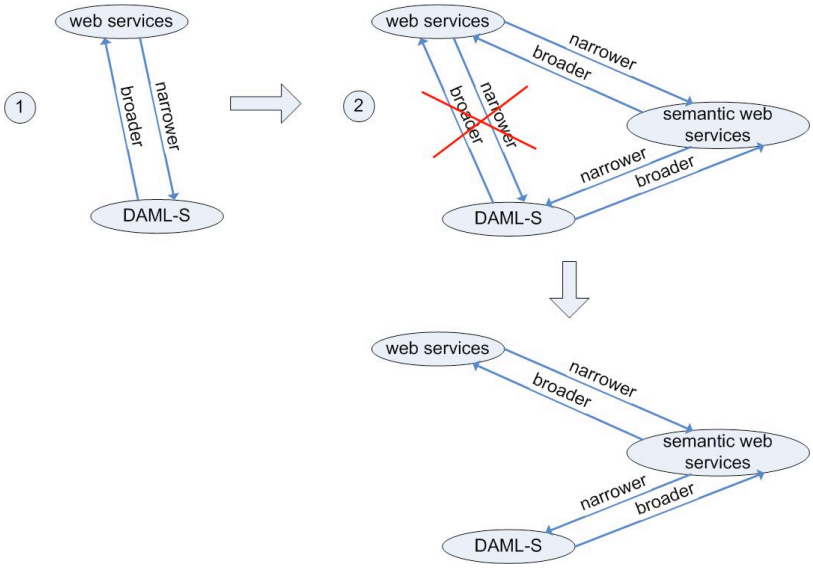

Figure 24. An algorithm applied when the created concept fits semantically in between two existing concepts.

As can be observed from Figure 24, firstly there is a concept "web services" with a narrower concept "DAML-S". When the concept "semantic web services" is created, the original relation between "web services" and "DAML-S" is deleted because it can be inferred from the relations with the new concept. Note that the same applies for multiple level relations. If a relation can be inferred from other relations with more intermediate concepts then the relation is removed.

8.4.2 Exploration task

There are only few notable improvements left concerning the exploration task, which have not been mentioned so far. As shown in Figure 25 and Figure 26 below, the tag vocabulary can be viewed in a plethora of different ways. Noteworthy is that we can

restrict the tag vocabulary to the tags we have entered to the system (see the options "show all tags" and "show my tags"). As well, the "tree" view option allows viewing the vocabulary as a conceptual tree, which we can easily explore using the semantic relations of the concepts (narrower, broader, and related concepts). By right clicking on a concept in the tree view a menu appears, which allows performing some actions. The most frequent action would be to view the bookmarks associated with the concept. Moreover, an important action is the "Merge with…" which allows merging a concept with one or more other concepts. The merging action generates a new concept, which includes the union of all the labels, the semantic relations, and the bookmarks of the merged concepts.

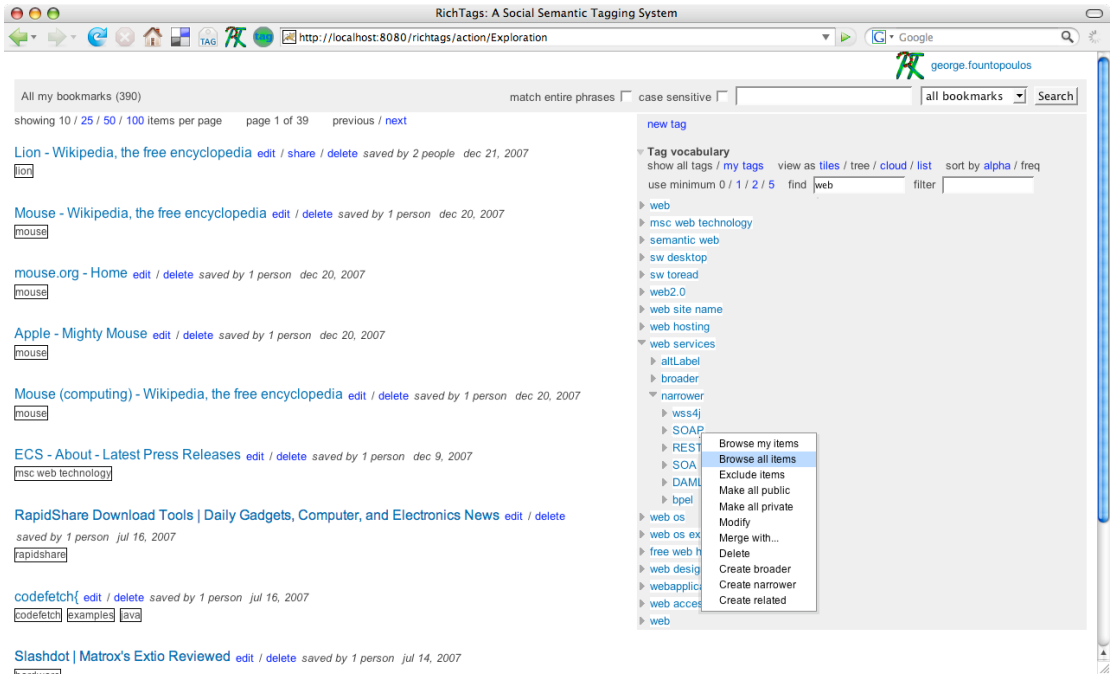

Figure 25. The tree view allows exploration of the conceptual hierarchy by viewing narrower, broader, or related concepts. Right clicking on a concept reveals a number of options associated with the concept.

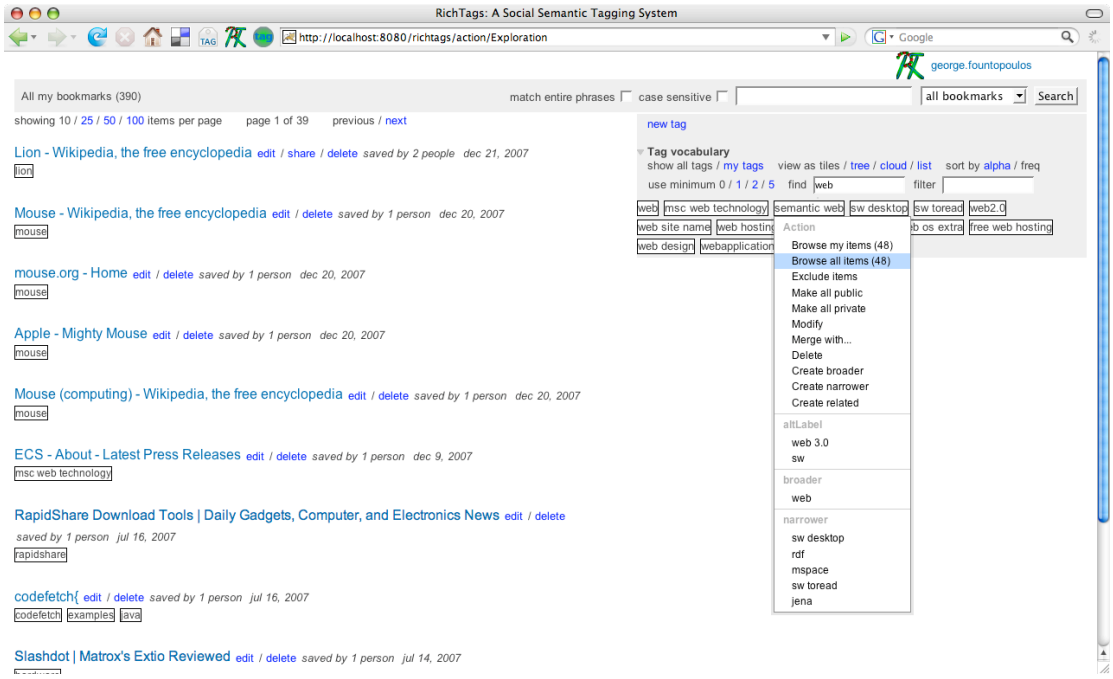

Figure 26. The tiles view. By clicking on a concept a number of options are presented along with the associated information for the concept (alternative labels, broader, narrower, and related concepts).

Finally, another interesting option is the ability to exclude items tagged with a particular concept as shown in Figure 27 below. As you can see, the search results for the concept "semantic web" can be restricted to those items not tagged with "semantic web to read list" choosing the option "Exclude items" of the concept.

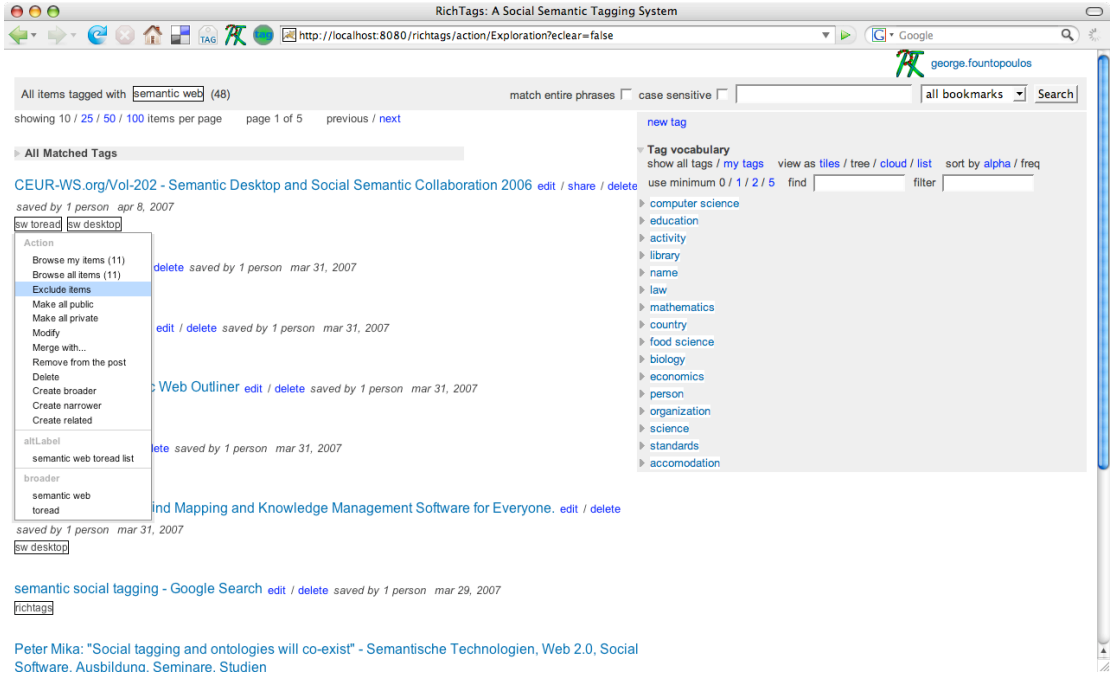

Figure 27. The "Exclude items" option allows excluding posts tagged with the particular concept.

9 Future work

Having presented the improvements of the RichTags social semantic tagging system over the current social tagging systems, here I will try to give future directions and present my perspective on what we could achieve in future.

9.1 Evaluation of social semantic tagging in use

Folksonomies have become a popular means for bookmarking with sites like Delicious [3] and Flickr [1] maintaining big communities of users. Ease of use is an important factor for success, and folksonomies can claim it since the tagging task just requires typing in some arbitrary keywords the user wants to attach for a resource. In contrast, the full potential of semantic tagging is achieved when the creation of a tag encompasses entering all the alternative labels and semantic relations for the tag, which obviously requires more effort. Though, note that it is not required to use multiple labels and semantic relations but the true value is added by doing so. Although the semantic tagging can be used as easily as the conventional tagging, the benefits appear when users specify relations and multiple labels for tags, which requires additional effort.

However, the extra effort for specifying semantic relations and multiple labels might not constitute a real implication since RichTags offers recommendations of semantic tags during the tagging task. A user will need to create a semantic tag only if no one else has created it before, which in case of popular tags will be reasonably rare.

9.2 New opportunities for content ranking

Social tagging systems offer additional opportunities for content ranking. As have been previously discussed, semantic tagging improves the relevance of the retrieved items but there are more factors beyond the relevance that could affect the order of the results. Even though semantic tagging offers 100% precision, which means that all the items are relevant, further concern is required to determine which of those relevant items would be the most preferable for the particular user. For example, a biologist will most likely prefer the items saved by one of his colleagues over those saved by a musician, no matter if both item sets are absolutely relevant. Furthermore, a user might want to explicitly specify the profiles of the users whose items he wants to retrieve from a given search query. As well, there might be people of particular

reputation whose items should be ranked more heavily. User profiling would be required to study such priority schemes.

9.3 Outlining a future Information Retrieval system

To attempt to define the ideal Information Retrieval system, I believe that, a future development would integrate all the basic Information Retrieval tools together into one unified environment, which would enable all the functionality in a consistent manner. Web 2.0 mashup technologies offer for such integration with Web Services APIs being available by most notable IR systems today (e.g. Google).

There is no reason for having separate tools for bookmarking and searching. An integrated environment would offer the benefits of both. Although it is known that Google and other search engines use the social bookmarking sites to improve their search results, however, to the best of my knowledge, none of them yet offers an integrated bookmarking service as a feasible substitution for all the social tagging tools[4].

Figure 28 below will help me to describe the search capabilities of what I currently perceive as the ideal Information Retrieval system. Note that the figure does not aim to present a good interface from the HCI perspective; rather it serves as a simplified demonstration of the required capabilities of the system. The exact software controls that should be used or the way they should be rendered is an HCI concern, and does not serve for this particular demonstration.

[4] Note that Google bookmarks (http://www.google.com/bookmarks/) and Yahoo! MyWeb2.0 (http://myweb.yahoo.com/) are not integrated with the corresponding search engines, and specifically Google bookmarks is limited to private posts hence is not a social bookmarking service.

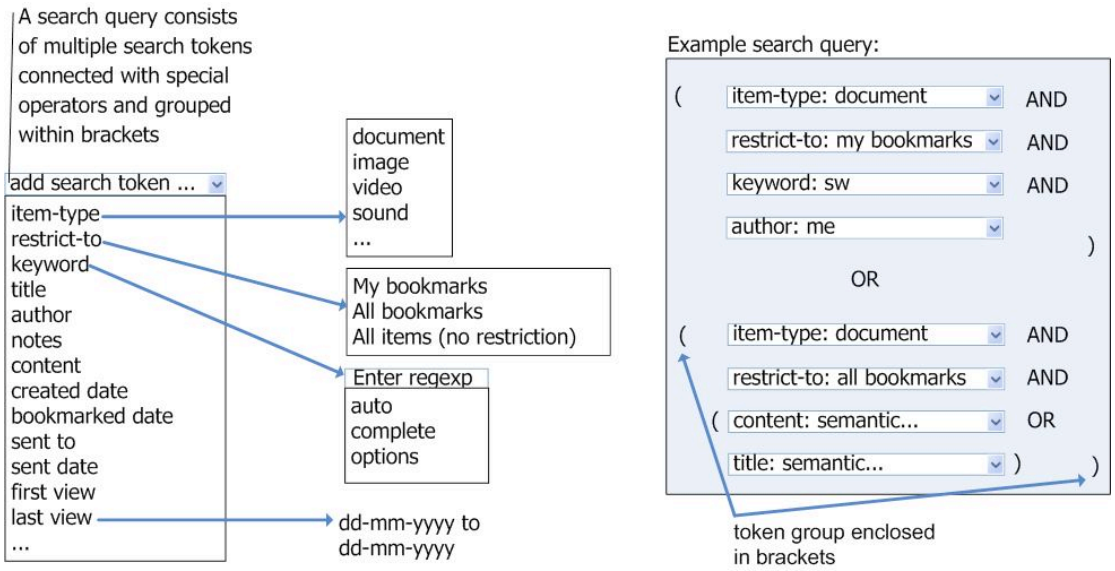

Figure 28. Search capabilities of a future Information Retrieval system.

As can be observed from the above Figure 28, the search query consists of several parts, which I call search tokens. Multiple search tokens can be connected with special operators and can be grouped within brackets to form more complex queries. Every search token is of a particular type, which indicates constraints over the allowed values for the token. Thus, the system should present a special interface for each token type, allowing the user easily selecting or entering a value within the range of valid values.

For example, the token type "item-type" allows selecting the kind of the items we want to retrieve (document, image, video, etc). Hence, when the user selects the token type "item-type", the system presents a fixed list of options from which the user can select an item type. On the other hand, the token type "keyword" refers to a keyword that has been associated with the item we are looking for (I use the term keyword as a synonym to tag and label). Thus, a different interface is used to enter the value, and other validation mechanism is applied. A reasonable approach for the keyword would be a text box with auto complete functionality, where the user can enter a regular expression that will be matched against keywords.

Depending on the value of the "item-type" token, the list of options for the subsequent search tokens is adjusted accordingly. Each item type has a specific set of fields applied to it. A document for example would include fields like author and title, whereas a video would include fields like duration and location. Thus, the options list

would be adjusted accordingly so to include only those fields applicable for the particular item type.

The "restrict-to" token type allows three fixed options. The first option (my bookmarks) means the results will include only those items the user has bookmarked. In fact this option supports the *recall task* I have been previously mentioned when giving an abstract definition of the problem. The second and the third options both are aspects of the *discovery task*. The "all bookmarks" option restricts to those items that have been bookmarked by the users of the system, and the "all items" option allows retrieving any item no matter whether it has been bookmarked or not. The later option is in fact equivalent to avoiding including the "restrict-to" token, and constitutes a typical search engine (like Google), which does not restrict the results to bookmarked items.

The token type "content" allows matching over the content of the items we are looking for (see content-based retrieval in Section 6.1). This is the kind of search that a typical search engine (like Google) currently offers. The value a user can enter for this token depends on the "item-type" token. For example, if the "item-type" is "document", then the value for "content" is some text, so the system presents a text box for the user to enter some text. In contrast, if the "item-type" is "video", then other mechanism should be applied to match against such content (see next Section for a relevant discussion).

Other token types, like the date types, allow specifying a range of dates, which, in the case of the "created date" token, restricts to those items created within the specified date range. The system would normally present a calendar control so to help the user easily specifying the range of dates.

9.4 Discussion

The token type "keyword" from the prior outline refers to a search based on keywords, which is what RichTags and other social tagging systems offer. As have been previously discussed, the defining characteristic of RichTags is that the tags are semantic, and this enables the system having all those advantages over the current tagging systems. The semantic keywords (or tags) in RichTags offer semantic search capabilities, comparing to the plain text matching offered by current systems.

However, keywords are not the only type of search token that can be applied to a search query. Other search tokens might include the type "content", which matches against the actual content of the items we are looking for. Such content might be text, image, sound, video, or any other type of content, which is specified by the "item-type" token. Thus, it is wise to think of applying the same principles for all of these different content types in order to enable semantic search capabilities based not only on keywords but also on the actual content. But, what would constitute *semantic content* for these different content formats? I will try to present a perspective on this.

In RichTags, what differentiates the semantic tags from the typical tags in a conventional system is the fact that every tag is uniquely identified and distinguished from others no matter if its properties are not unique. For example, two tags having the same labels are still distinguished from each other due to their unique ids. The later also enables the definition of semantic relations between the tags (narrower, broader, related, etc). Hence, to attempt to define the meaning of semantic content I suggest that:

Semantic content is the one that can be uniquely identified and distinguished no matter if its perceptible properties are not unique.

Let us consider what the above definition would mean for the different content types like text, picture, video, and sound.

Taking the text as an example content type, it is obvious that the visual representation of the words in a particular piece of text does not identify the meaning the words are carrying. The user needs to read the text in order to fit the words into a context. For example, simply by looking at the word "mouse" from a piece of text we cannot claim whether it refers to an animal or to a device. We firstly need to read the text in order to realise the exact meaning of the word. Moreover, a machine cannot identify the exact meaning of the word without applying a specific algorithm, even if iterating through all the words of the text. Thus, such text is not semantic, but what would constitute a semantic text? Consider a text where every word would have a unique id attached to it, which would uniquely identify the exact meaning the word is carrying. How we can implement a tool that will support convenient composition of such text is a separate concern. For now just imagine that as you type the words you are presented with a dictionary of definitions where you can select the exact definition for each

word you are typing. It is prominent that such text would constitute more value and would provide more possibilities for using it. We could apply semantic searching over the content in a similar manner as RichTags applies it for keywords (tags).

Respectively, semantic picture would mean a picture with metadata attached to it so to describe and identify the objects depicted in the picture. For example, my balcony's view might look identical to the view from my friend's house in Portugal (same trees, the sea, etc). But the one location is in Greece whereas the other is in Portugal. The two pictures depict objects that are visually the same but constitute separate things. Likewise, a picture of (say) three people does not identify them unless there are sufficient metadata, such as their names, their dates of birth, their origin, and so on. As semantic text constitutes more value, analogously, semantic pictures would provide more usage options, such as semantic searching over the content of the pictures.

In a similar manner, semantic video and semantic sound include sufficient metadata to allow semantic searching over their content. For example, a song has the lyrics associated with it so to enable semantic searching over the words of the song. MPEG-7 [32] is a multimedia content description standard that allows metadata to be associated with audio or video content in order to support efficient searching of that content. Thus, MPEG-7 can serve for making these content types semantic.

10 Conclusions

This writing introduced RichTags, which is a social semantic tagging system. RichTags aims to overcome some weaknesses of the conventional social tagging systems (folksonomies) by utilizing Semantic Web technologies. The defining characteristic of the system is that the tags constitute an ontology of meaningful concepts, which is collectively managed by the users of the system. Hence, the approach is called *social semantic tagging*. It overcomes the *polysemy*, the *synonymy*, and the *basic level* variation problems encountered in the conventional systems. As well, it offers higher *precision* and *recall*.

Positioning RichTags in the key design dimensions according to [1], it is a *free-for-all* system, with special rules applied for the collective management of the tag vocabulary. Moreover, RichTags is a *suggestive tagging* system, which means that

users are presented with suggested tags during the tagging task. A *bag-model* approach is used, since everyone's post for a given resource is saved and managed separately. Although RichTags at the moment is primarily focused on documents, there is no restriction on the resource type for the tagged items. Furthermore, RichTags does not force any particular source for the material to be tagged and no restrictions apply on the resource connectivity. Finally, no dedicated mechanism is currently provided to support social connectivity between users.

The RichTags web application design conforms to the Model-View-Controller (MVC) software design pattern [24]. A high level architecture consists of some client-side libraries (YUI library [25] and RichTags JavaScript library) and some server-side modules, such as the controller servlet, the JSP view, the business logic, and the Jena Semantic Web framework [26]. The Jena framework is used for the interactions with the ontology (part of the *model*). As well, a database is used to store all the users' preferences and other data used internally by the system. All the server-side components of the web application are deployed in a JSP/Servlet container.

The ontology, which holds the application's data, is available to third party applications in various forms and can be managed using the RichTags Web Service. The data can be either directly retrieved in raw OWL format, or queried in SPARQL, using the Joseki SPARQL engine [27], which is integrated into the RichTags web application. As well, the RichTags Web Service enables authenticated third parties to manipulate the ontology.

Current realization of semantic tagging basically concerns an effort to automatically derive semantics out of folksonomies without affecting the mechanism of tagging applied in them [1, 3, 18, 20, 21, and 22]. In contrast, RichTags's approach for semantic tagging is a social process relied on the collective intelligence of the users instead of automation methods. The later means that the users collectively expand the tag vocabulary throughout the tagging task, while consistency mechanisms are applied to keep the vocabulary consistent during this expansion.

The basic factor that differentiates RichTags from existing proposals for the enhancement of tags with meaning is that the primary mechanism relies on human collective intelligence and not on automation methods. However, this does not mean that the proposed automation techniques could not be combined with RichTags;

contrariwise they could be very useful to speed up the production of the initial set of semantic tags in the vocabulary. Nevertheless, I believe RichTags's approach for the enhancement of tags with meaning is superior, since automation methods cannot achieve the same accuracy as human intelligence can. Users are the ones who at the end of the day evaluate the usefulness of any system, and any machine-generated intelligence cannot compete with the collective intelligence of the actual users.

Another difference from existing proposals is that RichTags is not limited to enriching the tags with meaning; instead it utilizes this semantic information to improve the tagging and the exploration tasks of tagging systems.

Finally, future work should include the evaluation of social semantic tagging in use and the study of the new opportunities for content ranking derived from such systems. As well, in addition to the keyword-based retrieval, we should consider ways of applying RichTags principles for other kind of search, which will enable semantic search over different content types in future Information Retrieval systems.

11 References

[1] C. Marlow, M. Naaman, D. Boyd, M. Davis, *HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, To Read*, ACM Press, In proceedings of the 17th conference on Hypertext and Hypermedia (HT'06), pages 31–40, New York, NY, USA, August 2006.

[2] T. O'Reilly, *What is Web2.0. Design patterns and business models for the next generation of software*, O'Reilly Media, www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, 2005.

[3] S. A. Golder, B. A. Huberman, *The Structure of Collaborative Tagging Systems*, HP Labs technical report, 2005.

[4] B. Lund, T. Hammond, M. Flack, T. Hannay, *Social bookmarking tools (II): A Case Study – Connotea*, D-Lib Magazine, 11(4), April 2005.

[5] T. Hammond, T. Hannay, B. Lund, J. Scott, *Social Bookmarking Tools (I) – A General Overview*, D-Lib Magazine, 11(4), April 2005.

[6] *Open Directory Project*, http://www.dmoz.org.

[7] *Yahoo! Directory*, http://dir.yahoo.com.

[8] T. Vander Wal, *Folksonomy: Coinage and Definition*, http://vanderwal.net/folksonomy.html.

[9] D. E. Millard, M. Ross, *Web 2.0. Hypertext by any other Name?*, In proceedings of the 17th conference on Hypertext and Hypermedia (HT'06), Odense, Denmark, ACM Press, 27-30, 2006.

[10] T. Berners-Lee, *Weaving the Web*, Orion Business Books, London, 1999.

[11] T. Berners-Lee, J. Hendler, O. Lassila, *The Semantic Web*, Scientific American, 17 May 2001, pp. 34-43.

[12] O. Lassila, R. Swick, *Resource Description Framework (RDF) Model and Syntax Specification*, W3C Recommendation, 22 February 1999, http://www.w3.org/TR/REC-rdf-syntax/ (current 10 February 2004).

[13] D. Brickley, R. V. Guha, *Resource Description Framework (RDF) Schema Specification 1.0*, W3C Candidate Recommendation, 27 March 2000, http://www.w3.org/TR/rdf-schema/ (current 10 February 2004).

[14] M. Dean, G. Schreiber, *OWL Web Ontology Language Reference*, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/owl-ref.

[15] C. A. Goble, D. De Roure, *The grid: An application of the semantic web*, ACM SIGMOD Rec., vol. 31, pp. 65-70, 2002.

[16] D. De Roure, N. R. Jennings, N. Shadbolt, *The Semantic Grid: A Future e-Science Infrastructure*, Grid Computing: Making the Global Infrastructure a Reality, F. Berman, A.J.G. Hey, and G. Fox, eds., John Wiley & Sons, pp. 437–470, 2002.

[17] A. Ankolekar, M. Burstein, J. R. Hobbs, O. Lassila, D. McDermott, D. Martin, S. A. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, *DAML-S: Web Service Description for the Semantic Web*, In proceedings of the 1st international Semantic Web conference (ISWC 02), 2002.

[18] P. Heymann, H. Carcia-Molina, *Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems*, Stanford InfoLab Technical Report, 24 April 2006.

[19] A. Miles, D. Brickley, *SKOS Core Vocabulary Specification*, W3C Working Draft, http://www.w3.org/TR/swbp-skos-core-spec (current 10 May 2005).

[20] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. Tomlin, J. Zien, *SemTag and Seeker: bootstrapping the semantic web via automated semantic annotation*, In proceedings of the 12th international conference on World Wide Web, May 2003.

[21] L. Specia, E. Motta, *Integrating Folksonomies with the Semantic Web*, In proceedings of the 4th European Semantic Web conference (ESWC 2007), Innsbruck, Austria, Springer, 2007.

[22] C. V. Damme, M. Hepp, K. Siorpaes, *FolksOntology: An Integrated Approach for Turning Folksonomies into Ontologies*, In proceedings of the 4th European Semantic Web Conference (ESWC 2007), 2007.

[23] L. Page, S. Brin, R. Motwani, T. Winograd, *The PageRank citation ranking: Bringing order to the web*, Technical report, Stanford University, 1998.

[24] *Design Patterns: Model-View-Controller*, http://java.sun.com/blueprints/patterns/MVC.html.

[25] *The Yahoo! User Interface Library (YUI),* http://developer.yahoo.com/yui/.

[26] *Jena Semantic Web framework*, http://jena.sourceforge.net/.

[27] *Design Patterns: Data Access Object*, http://java.sun.com/blueprints/patterns/DAO.html.

[28] E. F. Codd, *A Relational Model of Data for large Shared Data Banks*, Communications of the ACM, vol. 13, no 6, June 1970.

[29] *MySQL Database server*, http://www.mysql.com/.

[30] *Joseki SPARQL engine*, http://www.joseki.org/.

[31] *Apache Axis2 Web Services engine*, http://ws.apache.org/axis2/.

[32] B. S. Manjunath, P. Salembier, T. Sikora, eds., *Introduction to MPEG-7: Multimedia Content Description Language*, John Wiley & Sons, 2002.