International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:7, 2008

Self-adaptation of Ontologies to Folksonomies in Semantic Web

Francisco Echarte, José Javier Astrain, Alberto Córdoba, and Jesús Villadangos

***Abstract*—**Ontologies and tagging systems are two different ways to organize the knowledge present in the current Web. In this paper we propose a simple method to model folksonomies, as tagging systems, with ontologies. We show the scalability of the method using real data sets. The modeling method is composed of a generic ontology that represents any folksonomy and an algorithm to transform the information contained in folksonomies to the generic ontology. The method allows representing folksonomies at any instant of time.

***Keywords*—** Folksonomies, Ontologies, OWL, Semantic Web.

I. Introduction

CAPTURING knowledge by using markup techniques and by supporting semantic annotations is a major technique for creating metadata. Currently in the WWW, folksonomies [1] are much extended as tagging system due to its ease of use, because they allow integrating heterogeneous resources, and users can collaborate in the resource tagging process. However, it is difficult to work on such information because it has not any structure and it is user dependent. On other side, Ontologies [2] provides a framework to handle structured information, and to extract conclusions from it. It is very interesting to provide mechanisms to turn existing syntactic resources into knowledge structures. Therefore, modeling folksonomies with ontologies is an important issue.

Folksonomies are totally dependent on users and their considerations about the contents, because they annotate contents directly and freely. Therefore, the main characteristic of Folksonomies is the absence of structure in the information classification, and this characteristic we consider is one of the main success factors of folksonomies. In fact, folksonomies have been applied in a number of social networks in Internet, helping to build the known as Web 2.0.

Ontologies provide a domain vocabulary and define, in different formalization levels, the semantic of the vocabulary and the structure that relates the terms of the vocabulary. Ontological engineering refers to the set of activities related with ontological process development, ontological life cycle, methods and methodologies to build ontologies and the set of tools and languages supporting ontologies. In the last years, ontologies have been the focus of attention in multiple fields of research like Knowledge Engineering and Artificial Intelligence; and they have been applied in many different areas like Knowledge Management, Natural Language Processing, eCommerce, Intelligence Systems Integration, Bioinformatics, Education, and in the emergence area of Semantic Web [3].

A Folksonomy is composed of text labels, called tags, and resources annotated with those tags. There is not any predefined hierarchy or restriction to define the tags; it relies completely upon user criteria. In the last years Folksonomies have been used in different social networks like Flickr[1] and del.icio.us[2], where users annotate images and links respectively. In these cases, users assign metadata (or capture knowledge) following a decentralized process.

The knowledge captured with folksonomies has some drawbacks similar to others present in other classification systems like taxonomies or thesauruses including polysemy, synonymy and granularity [4,5]. Another problem proper of Folksonomies is related with the semantic of the tags.

Users take into account the purpose of the tag to define it, and not only the information to be annotated. So, different tag types can be identified [4] depending on its purpose: (1) tags used to identify about what is the content and who is referred with the information, (2) tags with the function to indicate the thing which is annotated: blog, book, etc. (3) tags used to identify who is the author or the proprietary of the content, (4) tags used to create categories to simulate hierarchies, (5) tags used to identify characteristics of the content, usually adjectives (funny, bored, etc.) representing the personal view of the user who annotates the content, (6) tags used to represent the relation of the user with the content, as mythings, myjob, mycomments, etc. and (7) tags used to organize tasks, as toread, todo, search-work, etc.

The above problems refer to the annotation process. However, there are other problems related with the navigation, for searching and accessing contents. Navigation problems can be grouped in two blocks [6]: (1) reduced search capabilities; and (2) limited exploration.

Search capabilities are reduced due to linguistic and semantic limitations of tags. The searching results are limited to the tags used in the annotation process. Therefore, if a user

Manuscript received June 30th, 2008. This work was supported in part by the Spanish Government under Grant TIN2006-14738-C02-02.

F. Echarte is a PhD Student of the Universidad Pública de Navarra (e-mail: patxi@eslomas.com)

J. J. Astrain, A. Córdoba, and J. Villadangos are with the Dept. Ingeniería Matemática e Informática, Universidad Pública de Navarra, Campus de Arrosadía, 31006 Pamplona SPAIN (e-mail: {josej.astrain, alberto.cordoba, jesusv}@unavarra.es).

[1] http://www.flickr.com
[2] http://del.icio.us

has assigned the tag cat to a resource and other user is looking for animal, that resource will not be shown, or if the user searches resources related to tag television, those resources annotated with *tv* tag will not be shown.

In Folksonomies navigation is a very simple and intuitive process, which allows finding out interesting contents. There are two basic ways of navigation through the tag space: a) search and refine and b) visualization tools of the tag space, like the cloud of words. However, such mechanisms are not effective enough.

Search and refine are based on the selection or searching of a tag and the posterior refinement of the results. For example, when we look for the wordbook in del.ici.us we get a set of results similar to the illustrated in Figure 1. This figure shows the set of web pages annotated with the wordbook, and the set of related tags, which can be used to extend the search.

The related tags provide a very basic way to continue the searching process or to refine the search. For example, some related tags, which identify personal aspects like *to order* or *whishlist*, are not useful. So many times, it is better to access the returned links and to refine the search in them.

However, this navigation mechanism seems to be a good method of exploration, its utility decreases as the number of contents and tags increase [7]. The reasons, to be avoided in other developments, are the following: (1) tags are organized in alphabetic order without taking into account the relation between them; (2) low frequency tags are not showed, hiding some points of views of users about the information. So, one of the most interesting capacity of the Folksonomies (integrate diverse points of views) is reduced; (3) there are redundant tags, like *blog, blogs, blogging*, etc. that could be resumed; (4) personal tags are shown, like *toread*, which do not provide additional semantic information interesting for other users and (5) this visualization is provided at the first level only, and it is not used in other places.

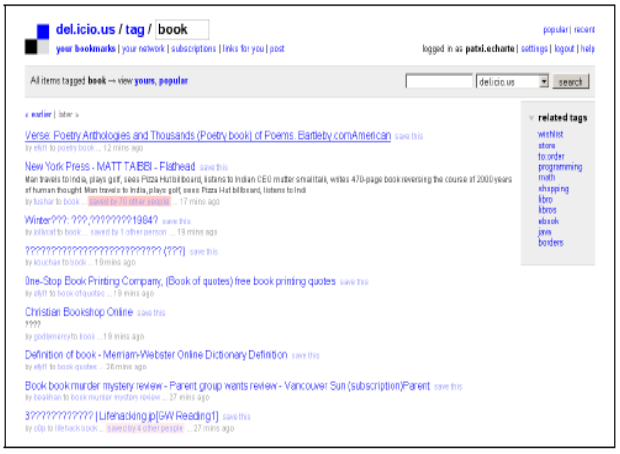

Fig. 1. Search and refine.

Gruber defined ontologies as the formal and explicit specification of a shared conceptualization [2]. In this definition, explicit refers to the requirement of name all concepts and elements of the domain; formal means that a formal language must be used; and shared means that the points of view of all users involved in the domain, are included in the ontology. Ontologies are structured knowledge where concepts, instances, attributes and relations are modeled. Thus, the ontology is a set of concepts which can be used by agents to dialogue with a common language, because the ontology models rigorously a domain.

There exist different languages used to define ontologies; however, it is important to use mature standards to define them. In this way, OWL [8,9] is the best option. It is proposed by the Wide Web Consortium (W3C) and is one of the main elements of the Semantic Web, a project initiated by Tim Berners-Lee [10] to include semantics in the current Web. OWL is supported by different technologies: XML, XML Schema, RDF, RDF Schema (the relation of OWL to XML and RDF is rather different), and all together provide a way to define a structure for the documents and to define the explicit semantic relation between different resources. All these technologies are open standards, tested and accepted. The semantic of ontologies in OWL is supported by the description logic formalism. Therefore, we have the axioms and inference rules necessary to derive conclusions from the information contained in the ontology.

A. Contributions

During the last year a significant body of publications deals with bringing folksonomies closer to ontologies ([11,12]). In [11] their representative replaces all the occurrences of tags in a certain group. In opposite, in this work we follow a method to model folksonomies, as tagging systems, with ontologies without modifiying the essence of the folksonomy. Annotations are associated to the tag variations using new elements (TagGroup). In [12] authors present a system, which provides a mechanism to organize resources by classifying the tags (or keywords) attached to them into predefined categories. The user is in charge of selecting ontologies which are relevant to the required categories. Concepts from these ontologies are used as categories. For example, to browse through the images of vehicles at Flickr, one would select a vehicle ontology. Once the ontology is pruned and refined, its concepts are used as categories.

Although there are many other projects [13,14,15] over ontologies to represent folksonomies, none of them estimate-the size of the ontology depending on the size of the folksonomy. Users cannot estimate the size of the ontology, and therefore he cannot decide whether it is worth using ontologies.

We propose a method to model folksonomies with ontologies. The method is composed of a generic ontology structure that represents any folksonomy; and an algorithm to integrate the information contained in the folksonomy with the generic ontology. Thus, we provide an algorithm to obtain an ontology that contains the tagged information from the folksonomy. The method improves, implements and extends the one proposed by Gruber in [18]. The user annotates the content, resources, documents, etc., and dynamically and without user intervention, the proposed modeling method stores the information in the ontology. This allows that the

extraction of the knowledge hidden in folksonomies using some of the features offered by ontologies and reasoners. This approach allows adapting ontologies as folksonomies evolve (users insert/delete/update annotations).

The method offers a basis for solving some folksonomies problems as: (i) deficiency of structure or relations among the existent tags [17]; (ii) tag variations (blog, blogs…); (iii) ineffective searches; (iv) tags definition based on the objective of the tag and not on the content (toread, whislist…) etc.

In addition, we provide a measure of the ontology size generated by our algorithm. This measure allows analyzing the system requirements before the transformation.

B. Paper Organization

The paper is organized as follows: section 2 introduces the modeling method and section 3 deals with the size estimation formula of the ontology. We prove the correctness (validity) of this formula and the method scalability, and finally, conclusions, future works and bibliographical references end the paper.

II. Modeling Method

Gruber in [18] does a presentation of the different roles of ontologies and folksonomies in Semantic Web, showing that both techniques are not completely opposed to each other, but is possible to get they complement each other. He presents some basic ideas on how to get it.

In his proposal, Gruber considers tagging, as the activity in which some user annotates some content, with one or more tags.

Tagging(object, tag, tagger)

Gruber also contemplates two more characteristics, centered first in the sharing of taggings between different sources or applications and one second oriented to indicate the polarity (positive or negative) of each tagging, in an attempt to reduce problems derived from spam or incorrect taggings.

Tagging(object, tag, tagger, source, + or -)

The work of Gruber presents the basic concepts involved in folksonomies, and offers a starting point for the creation of a method for modeling folksonomies with ontologies. However, in order to create this method, it is necessary to detail more explicitly the characteristics of folksonomies, and doing it in a stricter way, using some knowledge representation language like OWL DL, that offers the power and formalism of descriptive logics, and allows the expressivity needed to model the different characteristics.

A. Description

The ontology[3] has been designed in OWL language using Protégé tool [19]. This ontology defines the following classes: *Source, Resource, Tag, User, Annotation, AnnotationTag, Polarity*. These classes have the objective to represent the model of knowledge of folksonomies.

Source: as proposed by Gruber, represents the sources or applications that use or feed the folksonomy.*Resource:* represents any resource susceptible to be annotated. It has been renamed from object, as Gruber uses in his article, to get a broader meaning. This class could be specialized to represent documents, pictures, urls, or whatever other thing, in an application of the ontology to a concrete domain.

User: the objective of this class is to represent the users who do the tagging.

Tag: it represents the tag concept. This class has several properties related used to represent several tag variations, like syntactic variations, incorrect spellings, or even synonyms, with a unique instance. Two subclasses are also created: TagPersonal and TagCommon. Its objective is being able to classify the existing tags based on their type, separating those of personal type like related to planning of personal tasks or the self-reference tags (TagPersonal), of the rest of tags (TagCommon).

Annotation: it represents the action by which a user assigns a set of tags to a resource. Unlike the proposal of Gruber, who bases each annotation on the relation of a tag with an object, this class represents better the habitual behaviour of users, consisting of assigning several tags to the annotated resources.

AnnotationTag: since the Annotation class represents a set of assigned tags to a resource, this new class has been created to allow relating each annotation to the assigned tags.

Polarity: this class represents the polarity of each annotation, negative or positive. This polarity is associated to the instances of *AnnotationTag*, so it is possible to represent the assignation of several tags to a resource, and the polarity associated to each one of the assigned tags.

The ontology also contains a series of properties. In the following table (Table I) these properties are described indicating their name, domain and range.

TABLE I
Ontology Properties

Property	Domain	Range
hasPrefLabel	Tag	String
hasAltLabel	Tag	String
hasHiddenLabel	Tag	String
hasRelatedResource	Tag	Resource
hasSourceName	Source	String
hasUserName	User	String
hasURI	Resource	String
hasSource	Annotation	Source
hasUser	Annotation	User
hasResource	Annotation	Resource
hasDateTime	Annotation	dateTime
hasAnnotationTag	Annotation	AnnotationTag
hasTag	AnnotationTag	Tag
hasLabel	AnnotationTag	String
hasPolarity	AnnotationTag	Polarity
hasPosition	AnnotationTag	int

It is necessary to explain more in depth some of these properties, specially *hasAltLabel*, *hasHiddenLabel* and last, *hasPosition*. With respect to the two first, *hasAltLabel* and *hasHiddenLabel*, their objective is to represent the different variations of a tag, including singular and plurals, verbal tenses, synonyms, misspellings, incorrect syntactic forms, etc., from the tag's preferred representation. For example, the tag with preferred value semantic-web, could have associated to *hasAltLabel* the strings *semantics-web*, *semanticweb*, and to

[3]http://www.eslomas.com/tagontology-1.owl

hasHiddenLabel the strings *seantic-web*, *semantic wev*, etc.

With respect to *hasPosition* property, the convenience of this property is determined by the existence of works that indicate that when a user assigns a series of tags to a resource, the order in which he does it, is not accidental, and that different annotations made by different users, agree more frequently in the first tags that in the last ones [4]. This characteristic can be used for example to help in the creation of taxonomical structures over tags.

To represent the implicit order of the different tags in an annotation, the most convenient solution would have been to define each Annotation like an ordered set of *AnnotationTag* instances; however, this is not possible with the actual specification of OWL. The reason is the inexistence of any explicit mechanism in OWL to define ordered lists of elements. Although some works in the bibliography represent sequences of elements as linked lists [20], the solution based in *hasPosition* property has been adopted due to its simplicity and efficiency, allowing getting this information directly.

The considered approach produces an ordination which is not directly accessible from an OWL based model using a reasoner, but simplifies the common requirements of recovering the tags assigned to a certain resource, or the resources tagged with a certain tag, using SPARQL [21] queries for such purpose.

The proposed ontology is completed with a set of restrictions (see Table II) applied to the classes and properties described, that aid to represent the knowledge contained in the folksonomy, and allow validating the information generated in the ontology.

The described ontology makes possible to have a complete

TABLE II
RESTRICTIONS

Class	Restrictions
Source	Cardinality(hasSourceName) = 1
Resource	Cardinality (hasURI) = 1
User	Cardinality (hasUserName) = 1
Tag	Cardinality (hasPrefLabel) = 1
Annotation	Cardinality (hasSource) = 1
	Cardinality (hasResource) = 1
	Cardinality (hasUser) = 1
	Cardinality (hasDateTime) = 1
	Cardinality (hasAnnotationTag) >= 1

representation of any folksonomy. However, it is necessary to specify an algorithm that allows transforming existing folksonomies to the described ontology. On the other hand it is necessary to consider that a folksonomy is not a static classification system, but that evolves as the users create new annotations on resources, and that these annotations must be incorporated to the ontology. With the purpose of solving both situations, the transformation of a folksonomy to the described ontology, and its evolution in the time, an algorithm is proposed. This algorithm represents the set of actions necessary to model the annotations made by the users, producing the creation of a set of elements in the ontology.

The code (Fig. 2) shows the kernel of algorithm used to transform any annotation made by a user to the proposed ontology.

```
- PRE: User and Source previously defined

If not exists Resource then
        Create Resource
        Resource.hasURI = uri
End If

Create Annotation
Annotation.hasDateTime = annotation date
Annotation.hasSource   = Source
Annotation.hasResource = Resource
Annotation.hasUser     = User

For Each tag assigned to the resource
        If not exists Tag with the same prefLabel or altLabel or hiddenLabel then
                Create Tag
                Tag.prefLabel = text of the assigned tag
        End If
        Create AnnotationTag
        AnnotationTag.hasLabel = text of the assigned tag
        AnnotationTag.Tag      = Tag
        AnnotationTag.Position = position of the tag in the set of assigned tags
        AnnotationTag.Polarity = [Positive|Negative]

        Tag.hasRelatedResource += Resource
        Annotation.hasAnnotationTag += Annotation
End For
```

Fig. 2. Transformation Algorithm.

Table III shows an example of annotation in which user identified by 61baaeba8de136d9c1aa9c18ec3860e8 annotates the article 42 at date 2004-11-04 02:25:05 with 3 different tags: *metabolism*, *barabasi* and *networks*. The algorithm transforms this annotation into the corresponding OWL code (Fig. 3).

```
:usr_217369743f9df99964fal6439a01f5f rdf:type :User ;
        :hasUserName "217369743f9df99964fal6439a01f5f"^^xsd:string .
:rsrc_42 rdf:type :Resource ;
        :hasURI "42"^^xsd:string .
:AnnotationTag_47b094813cec3 rdf:type :AnnotationTag ;
        :hasPosition "1"^^xsd:int ;
        :hasLabel "metabolism"^^xsd:string ;
        :hasPolarity :Positive ;
        :hasTag :tag_metabolism .
:AnnotationTag_47b094813de62 rdf:type :AnnotationTag ;
        :hasPosition "2"^^xsd:int ;
        :hasLabel "systems_biology"^^xsd:string ;
        :hasPolarity :Positive ;
        :hasTag :tag_systems_biology .
:AnnotationTag_47b094813ee02 rdf:type :AnnotationTag ;
        :hasPosition "3"^^xsd:int ;
        :hasLabel "networks"^^xsd:string ;
        :hasPolarity :Positive ;
        :hasTag :tag_networks .
:Annotation_47b094813bf23 rdf:type :Annotation ;
        :hasDateTime "2006-12-26 02:03:08"^^xsd:dateTime ;
        :hasAnnotationTag
                :AnnotationTag_47b094813cec3 ,
                :AnnotationTag_47b094813de62 ,
                :AnnotationTag_47b094813ee02 ;
        :hasResource :rsrc_42 ;
        :hasSource :src_citeulike ;
        :hasUser :usr_217369743f9df99964fal6439a01f5f .
:tag_metabolism rdf:type :Tag ;
        :hasPrefLabel "metabolism"^^xsd:string ;
        :hasRelatedResource :rsrc_42 .
:tag_networks rdf:type :Tag ;
      :hasPrefLabel "networks"^^xsd:string ;
      :hasRelatedResource :rsrc_42 .
:tag_systems_biology rdf:type :Tag ;
        :hasPrefLabel "systems_biology"^^xsd:string ;
        :hasRelatedResource :rsrc_42 .
```

Fig. 3. OWL code.

TABLE III
ORIGINAL ANNOTATION

Resource	User	Date	Tag
42	61baaeba8de136 d9c1aa9c18ec38 60e8	2004-11-04 02:25:05.373798 +00	metabolism
42	61baaeba8de136 d9c1aa9c18ec38 60e8	2004-11-04 02:25:05.373798 +00	barabasi
42	61baaeba8de136 d9c1aa9c18ec38 60e8	2004-11-04 02:25:05.373798 +00	networks

III. ONTOLOGY SIZE ESTIMATION ANALYSIS

With the purpose of being able to make comparisons with other models based on ontologies, it is necessary to have some type of measurement to evaluate the amount of resources used.

This size will be function of the existing instances at every moment in the modeled folksonomy, formed by the number of sources, users, resources, tags and annotations. Since the information represented in ontologies based in OWL is stored in form of RDF triplets or statements, the number of triplets represented by ontology can be considered a good factor of measurement of the necessary resources.

The size in number of triplets, of the resultant ontology, could be estimated by the following formula:

$$\text{Number of triplets} = (2s + 2u + 2r + 2t + 5a) + (6\alpha a + \beta r) \quad (1)$$

Being *s* the number of sources, *u* the number of users, *r* the number of resources, *t* the number of tags, *a* the number of annotations, α the average number of tags per annotations, and β the average number of tags assigned to a resource. These formula is based on the fact that the method creates two triplets for each instance (*rdf:type* and one property) of *Source*, *User*, *Resource* and *Tag*, five triplets for each instance of *Annotation* (*rdf:type* and four properties), six triplets for each instance of *AnnotationTag* (*rdf:type*, four properties and the relation with an *Annotation* instance), where the total number of *AnnotationTag* instances are the number of *Annotation* instances multiplied by the average number of tags per annotation (α), and finally, one triplet (*hasRelatedResource* property) for each resource-tag relation (βr).

These values (α and β) depend on several things, such as the possibilities of the user interface where the annotations are created, the type of users, or the concrete field where the folksonomy is being applied. Several works exist in the bibliography [4,22,23] or in web sites of some of this applications, as del.icio.us, in which some of this data can be studied.

A. Workbench

To test the validity of the estimation formula and scalability of the method, we collect data from the social web CiteULike[4] (which contains bibliographic cites) collecting a total number of 2,290,740 annotations[5]. Each annotation consists on a tag assigned by a user to a resource, at a given date. In this way one annotation with several tags, would be represented by one registry for each tag assigned to the resource. After a first analysis of the data set, the existences of two tags with a significantly larger number of annotations than the rest were detected. This could be interpreted as if they would be created by some automatic procedure. These tags were *bibtex*-import, with 178,813 annotations and no-tag, with 73,755 annotations. Once deleted the annotations associated to these tags, the resulting data set had the following characteristics:

- Number of annotations: 2,038,172, one record per *user-tag-resource*
- Number of resources: 494,206
- Number of users: 21,480
- Number of tags: 151,522

This information has been implemented on a MySQL database. The average number of tags per annotation and the average number of tags per resource are calculated based on these data. Note that in the data set there is one record for user-*tag-resource*. This means that each annotation is represented with several records, one for each tag. Therefore is necessary to calculate the total number of annotations. Results obtained show 580,295 annotations, so, the average number of tags per annotation is 3.512. This number is calculated with the following SQL sentence:

```
SELECT COUNT(DISTINCT(CONCAT(artId,'#',user)))
FROM annotations;
```

In the same way, the total number of tag-resource assignations is calculated with the following SQL sentence:

```
SELECT COUNT(DISTINCT(CONCAT(artId,'#',tag)))
FROM annotations;
```

As result the total amount of assignations is 1,964,721, providing an average number of 3.98 tags assigned to resource. To perform the tests, the whole data set has been split in 10 subsets ($data\_subset_1$ to $data\_subset_{10}$). Each subset contains 50,000 resources with theirs annotations, associated tags and user's definitions. Each one of the subsets has been transformed following the proposed transformation algorithm and implemented in SDB [24], a RDF triplet store. Load time, index generation time, the number of triplets, and the performance of some queries, have been measured.

The queries have been performed with SPARQL sentences using the RDF store managed by SDB, which allows to use different SQL databases and different table layouts. The selected database has been MySQL 5 (with default configuration) and the layout2/index layout.

The experimental environment used consists on a 2.13GHz Xeon 3050 processor Linux server with 2GB RAM and two RAID1 (software) SATA hard disks (250GB). Initially,

[4] http://www.citeulike.org

[5] http://www.eslomas.com/index.php/publicaciones/tagontology

several data stores (*citeulike_sdb*$_1$ to *citeulike_sdb*$_{10}$) have been defined in order to perform the experimental tests on the modeled folksonomy, where:

$$citeulike\_sdb_n = \bigcup_{i=1}^{n} data\_subset_i \text{, being n = 10.}$$

The script used to perform the tests builds each one of the *citeulike_sdb*$_n$ and evaluates the previously defined parameters. The script performs also the evaluation of four different SPARQL queries, common in folksonomies.

```
Query 1 Tags associated to a resource
PREFIX to: <http://www.eslomas.com/tagontology-1.owl#>
SELECT distinct ?t WHERE{
? t to:hasRelatedResource to:rsrc_420639 }

Query 2 Resources associated to a tag
PREFIX to:<http://www.eslomas.com/tagontology-1.owl#>
SELECT distinct ?r WHERE {
to:tag_bioalgorithms to:hasRelatedResource ?r }

Query 3 Tags used by a user
PREFIX to: <http://www.eslomas.com/tagontology-1.owl#>
SELECT distinct ?t WHERE {
?a to:hasUser to:usr_ee4410113b1c0d182ccf1762ceede49 .
?a to:hasAnnotationTag ?at.
?at to:hasTag ?t }

Query 4 Resources annotated by a user
PREFIX to: <http://www.eslomas.com/tagontology-1.owl#>
SELECT distinct ?r WHERE {
?a to:hasUser to:usr_ee4410113b1c0d182ccf1762ceede49 .
?a to:hasResource ?r }
```

Figure 4 shows the number of triplets for each store *citeulike_sdb*$_n$. The number increases linearly with the size of the store. The load and index generation times are analyzed in Figure 5.

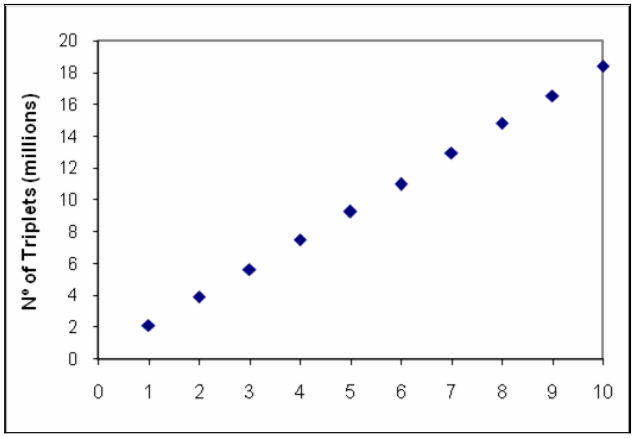

Fig. 4. Number of triples (in millions) per store.

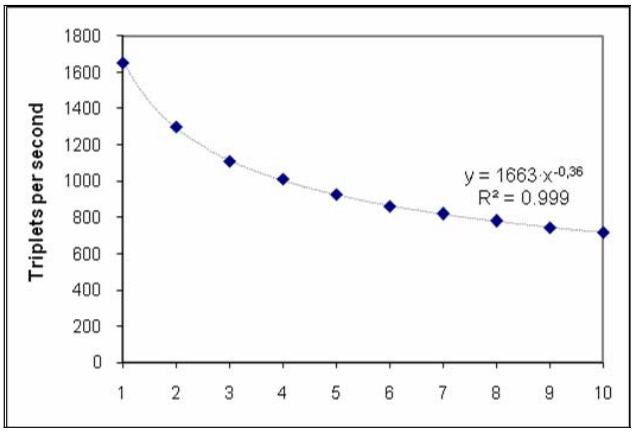

Fig. 5. Load and index generation time.

Figure 6 shows the number of triplets per second loaded in the stores. Initially, the speed decreases considerably with the data store dimension. After the third subset loaded this decrease becomes more moderated. The speed seems to stabilize near 600-800 triplets per second.

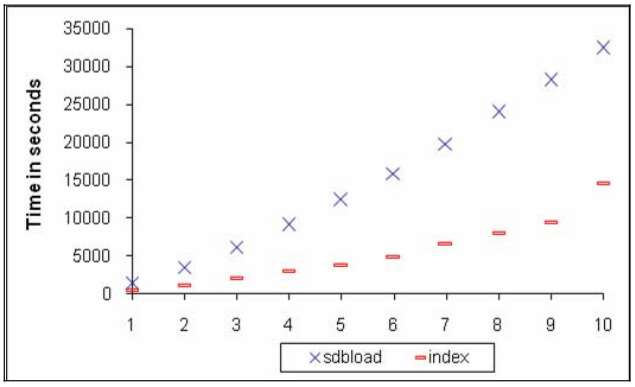

Fig. 6. Triplets per second stored in the repositories.

Figure 7 shows the number of triplets in the stores versus the estimation given by (1), validating the proposed formula. Deviation between estimations and real values is very close to zero as is shown in Table IV.

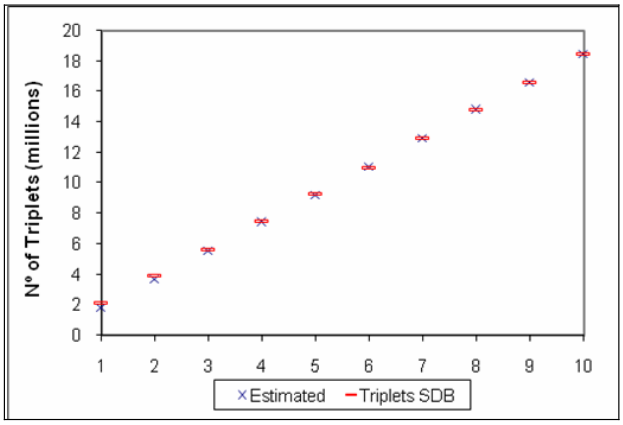

Fig. 7. Formula validation.

Figure 8 shows the number of triplets in the store versus the estimation given by (1), validating the proposed formula.

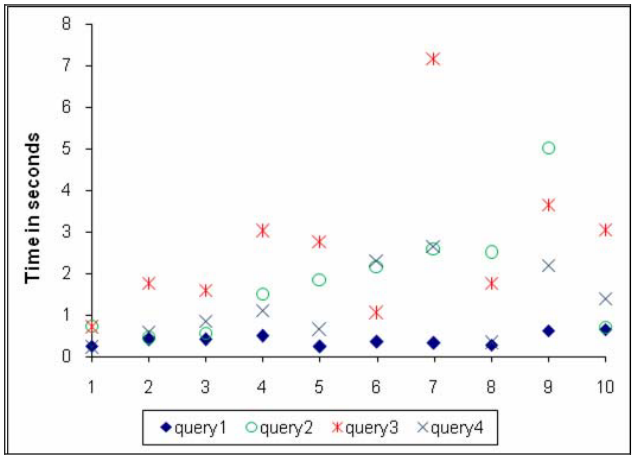

Fig. 8. Response time analysis for different queries.

The above figure (Fig. 8) shows that queries response time has a higher dependency on the specific searched data than in the total volume of information in the store. It shows the response time of the four described queries over each *citeulike_sdb*$_n$ store. Query 3 has a high average response time for store *citeulike_sdb*$_7$. The reason is that one of the searched users has a very high number of annotations: 1,176 versus an average of 27 in the rest of users and an average number of tags per annotation of 16.7 versus the total average value of 3.98.

In conclusion, the experimental results are:

1. The ontology size increases linearly with the number of elements in the folksonomy.
2. The average response time to perform a set of typical queries in folksonomies increases linearly and it has a higher dependency on the specific searched data than in the total volume of information.
3. The user annotates the content, resources, documents, etc., and dynamically and without user intervention, the proposed modeling algorithm stores the information in the ontology.

IV. Conclusions

A method for modeling folksonomies with ontologies has been proposed. The modeling method is composed of a generic ontology that represents any folsonomy and an algorithm to transform the information contained in folsonomies to the generic ontology. The method allows the self-adaptation of the ontology as the folksonomy evolves.

The method improves, implements and extends the one proposed by Gruber. We validate the proposed formula to estimate ontology size and the scalability of the method.

V. Future Works

The usefulness of the SPARQL queries in large and complex real systems must be analyzed. In these systems, users query for more than one tag, hence, “self-joins” are necessary. In addition, users would like to see more than just a plain resource identifier. This fact makes joins necessary with additionally metadata (e.g., the tags the user had assigned to the resource). In these systems, queries would be much more complex.

Acknowledgements

This work is partially supported by the Spanish Research Council under research grant TIN2006-14738-C02-02.

References

[1] Vander Wal. T.: Folksonomy. http://vanderwal.net/folksonomy.html, February 2 (2007)
[2] Gruber, T. A.: Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2), 199--220 (1993)
[3] Gómez-Pérez, A, Fernández-López, M., Corcho, O.: Ontological Engineering, London, Springer-Verlag (2003)
[4] Golder, S.A., Huberman, B.A.: The Structure of Collaborative Tagging Systems. Journal of Information Science 32, 2, 198--208 (2005)
[5] Brickley D. and Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February (2004)
[6] Begelman, G., Keller P., Smadja, F.: Automated Tag Clustering: Improving search and exploration in the tag space. WWW2006, May 22-26 (2006) Edinburgh, U.K.
[7] Mathes, A. Folksonomies - Cooperative Classification and Communication Through Shared Metadata. Computer Mediated Communication, Dec (2004)
[8] Smith, M.K., Welty, C., McGuinness, D.L.: OWL Web Ontology Language Guide. W3C Recommendation 10 February (2004)
[9] Dean, M. and Schreiber, G.: OWL Web Ontology Language Reference. W3C Recommendation 10 February (2004)
[10] Berners-Lee, T., Hendler, J. and Lassila O.: The Semantic Web. Scientific American, May (2001)
[11] Specia, L., Motta, E.: Integrating Folksonomies with the Semantic Web. In: European Semantic Web Conference (ESWC 2007), The Semantic Web: Research and Applications. LNCS 4519, pp. 503—517, Springer. Heidelberg (2007)
[12] Abbasi I R, Staab S., Cimiano P: Organizing Resources on Tagging Systems using T-ORG. In: Bridging the Gap between Semantic Web and Web 2.0, workshop at European Semantic Web Conference (ESWC 2007), Heidelberg (2007)
[13] Robustai. http://robustai.net/folksonomy/Tag-ontology.html
[14] tagont. http://code.google.com/p/tagont/
[15] Tag ontology design. http://www.holygoat.co.uk/projects/tags/
[16] NEPOMUK. Annotation ontology specification. http://www.www.semanticdesktop.org/ontologies/nao/
[17] Passant, A.: Using Ontologies to Strengthen Folksonomies and Enrich Information Retrieval in Weblogs. International Conference on Weblogs and Social Media, ICWM. March (2007)
[18] Gruber, T.: Ontology of Folksonomy: A Mash-up of Apples and Oranges. AIS SIGSEMIS Bulletin, 2(3&4) (2005)
[19] Protégé, http://protege.stanford.edu/
[20] Drummond, N., Rector, A., Stevens, R., Moulton, G., Horridge, M., Wang, H.H., Seidenberg, J.: Putting OWL in Order: Patterns for Sequences in OWL. OWLed (2006)
[21] Prud'hommeaux, E., Seaborne A.: SPARQL Query Language for RDF. W3C Working Draft 4 October (2006)
[22] Damianos, L. E., Cuomo, D., Griffith, J., Hirst, D. M., Smallwood, J.: Exploring the Adoption, Utility, and Social Influences of Social Bookmarking in a Corporate Environment. In 40th Hawaii International Conference on Systems Sciences (2007)
[23] Brooks, C.H. and Montanez, N.: Improved Annotation of the Blogosphere via Autotagging and Hierarchical Clustering. WWW 2006, May 23--2 (2006), Edinburgh, UK.
[24] Sdb – jena. http://jena.hpl.hp.com/wiki/sdb