Chapter 9
Semantic Models for Interactive Systems: The Case of Tagging and Folksonomies

Steffen Lohmann

Abstract Tagging, i.e. the annotation of resources with arbitrary text labels by users, has become a popular indexing method for interactive systems in the last few years. The linked vocabulary resulting from tagging is known as folksonomy and provides a valuable source for the exploration of digital resources. However, the interoperable use of folksonomies and related user interface components requires a consistent and comprehensive domain description. For this purpose, we developed a semantic model that describes the main concepts and relationships in the domain of tagging in a consistent and extensible way. It contributes to a better domain understanding and facilitates the development of interactive systems that use tagging as indexing method. By using the semantic model, folksonomies become independent from individual systems, which increases their interoperability and the reusability of related user interface components.

9.1 Introduction

Having its roots in social bookmarking and media sharing, *tagging* has become a popular indexing method in the last few years and can now be found in many interactive systems. In this indexing method, users annotate digital resources with arbitrary text labels, so-called *tags*, in order to organize the resources for themselves and/or others. What is considered a resource depends on the application context. It can be a web page *tagged* with social bookmarking services like Delicious, a photo or video on media sharing websites like Flickr or YouTube, or a mail in an email client. Even digital references of physical objects can be tagged, as long as they are uniquely addressable. For instance, books that are referenced in cataloging websites like LibraryThing or products in online shops like the one of Amazon are also subject to tagging.

In contrast to other keyword-based indexing methods, the people who perform the tagging are not professionals (e.g., authors, publishers, librarians, etc.) but common users. Furthermore, tagging breaks radically with most traditional forms of

S. Lohmann (✉)
University of Stuttgart, Stuttgart, Germany
e-mail: steffen.lohmann@vis.uni-stuttgart.de

T. Hussein et al. (eds.), *Semantic Models for Adaptive Interactive Systems*,
Human–Computer Interaction Series, DOI 10.1007/978-1-4471-5301-6_9,

indexing by using neither a controlled vocabulary nor a hierarchical structure for classification. Instead, a tag can be any character string a user considers helpful in organizing a resource. Even though many interactive systems recommend tags, no terms are ‘forced’ onto users but people are free to use their own vocabulary. This vocabulary of the people along with the many links resulting from tagging has come to be called *folksonomy*.

Tagging and folksonomies have also become popular research topics in the last few years. They have been analyzed and utilized in a number of works, resulting in several interesting findings, for example, on tag use and distribution (Peters 2009). However, they still lack a shared understanding and common conceptualization. Though several models and representations have been proposed (Kim et al. 2008; Lohmann et al. 2011), a consistent and comprehensive description of tagging is still missing. Such a consistent domain description is not only important for a better understanding of tagging and folksonomies but can also improve the interoperability and reusability of interactive systems that use tagging as indexing method. Especially reusable user interface components require a conceptual representation that is independent from individual systems.

In order to close this gap, we developed a semantic model that describes the main concepts and relationships in the domain of tagging. In the following, we present this model and illustrate its benefits for interactive systems. We start with a description of the core concepts and relationships in the domain of tagging and folksonomies in Sect. 9.2. In Sect. 9.4, we present an ontology that implements our semantic model. We illustrate its application by an example scenario in Sect. 9.5 and show how graph visualizations can be derived from it in Sect. 9.6. In Sect. 9.3, we discuss related work before we conclude this chapter in Sect. 9.7.

9.2 Concepts and Relationships

In order to create a semantic model for tagging and folksonomies, we first need to identify the core concepts and relationships in the domain. Apart from basic structures and elements, we must also consider related and more advanced concepts.

9.2.1 Basic Concepts and Relationships

Tagging consists of three sets of elements that form the basis for the semantic model (Mika 2005; Heymann and Garcia-Molina 2006; Smith 2008; Peters 2009, p. 157):

1. *Resources* that are being tagged. As mentioned in the introduction, these resources can be anything, as long as they are uniquely addressable by the interactive system.
2. *Users* who perform the tagging. In tagging contexts, the term ‘user’ denotes all people who use an interactive system for tagging, independently of their role and motivation.

3. *Tags* that are associated with the resources. Tags can be common words, slang, abbreviations, emoticons, star-ratings, or even individual text strings that are only meaningful to the person who assigns them, but not for others.

Though these elements are differently named in the literature, their semantics and relationships are always the same: One or more *users* (or people, actors, etc.) annotate *resources* (or objects, instances, etc.) with one or more *tags* (or keywords, labels, etc.). This fundamental principle of tagging can be defined as an axiom, as it has to be true for any folksonomy:

Axiom 1 *Each tagging links exactly one resource with one user account and one or more tags.*

Apart from that, there are some additional principles that are key to tagging and that can also be defined as axioms:

Axiom 2 *Each tag can be assigned at most once to each resource by each user account.*

Axiom 3 *A tag has always exactly one label—otherwise it is not a tag.*

More formally, a semantic model for tagging and folksonomies consists of three finite and disjoint sets $R = \{r_1, r_2, \ldots, r_k\}$, $T = \{t_1, t_2, \ldots, t_l\}$, and $U = \{u_1, u_2, \ldots, u_m\}$ that represent the *resources*, *tags*, and *users*. They are interconnected by taggings, i.e. the set of annotations $A = \{a_1, a_2, \ldots, a_n\}$ that defines the relationships according to the axioms given above. A basic model for folksonomies can thus be described by the quadruple $F = (R, T, U, A)$ (Hotho et al. 2006; Lohmann and Díaz 2012).

9.2.2 Further Concepts and Relationships

However, there is more than the interlinked resources, tags, and users that must be considered in a comprehensive description of the domain of tagging and folksonomies. Another important piece of information is the date and time of tagging. Many interactive systems use this information to display taggings in reverse chronological order, while others enable users to define time intervals when browsing folksonomies (Li et al. 2007). This is why some consider *time* as another core element of folksonomies (Wu et al. 2006; Smith 2008, p. 101).

Others emphasize the source of tagging as an important piece of information (Gruber 2007). This is particularly true with regard to the interoperability of folksonomies: Source information is important if folksonomies leave the borders of one interactive system, for example, to be shared with other systems and/or merged with other folksonomies. In these cases, it may be relevant to know which parts of the folksonomy are from which system.

Examples of further concepts that need to be taken into account are comments added to taggings or hierarchical tag relations, as they can be defined in some interactive systems (e.g. Bibsonomy). Likewise, there can also be links between resources (e.g. hyperlinks) or users (e.g. group links). Though these relationships are not part of the folksonomy itself, they must be taken into account by the conceptualization.

A comprehensive semantic model must also consider more advanced forms of tagging. For instance, some systems (e.g. Flickr or Bibsonomy) support *group tagging* by enabling the creation of a group account that single user accounts can be linked to. Other systems (e.g. Faviki) offer features for *semantic tagging*, where the meaning of tags is disambiguated by linking them to well-defined entities, such as DBpedia resources (Bizer et al. 2009) or Wordnet terms (Miller 1995). Finally, *automatic tagging* denotes tagging with automatic tags, i.e. text labels that are automatically assigned to a resource by the interactive system. Strictly speaking, the latter is not really tagging, as there is no user involved. However, since automatic tagging is an important concept of the domain, it should also be taken into account by the semantic model.

A powerful way to formally describe all this information is an *ontology*. In computer science, an ontology is briefly defined as "an explicit specification of a conceptualization" (Gruber 1993). Ontologies gained much popularity with the rise of the Semantic Web as a way to give information well-defined meaning. They describe the concepts and relationships of a semantic model in a logic-based language that allows for machine interpretation and automated reasoning.

9.3 Existing Ontologies

Before starting to develop a new ontology, it is recommended to look for existing ontologies that describe the same or a similar domain and examine if they can reasonably be reused (Noy and McGuinness 2001). In case of tagging and folksonomies, there already exist several ontologies.

Early conceptualizations have been presented by Gruber (2007, 2005), Newman (2005) and Mika (2005). They do a good job in describing the basic concepts and relationships of tagging and folksonomies, as they were defined above. While Gruber's conceptualization is a rather informal description of ideas and Mika's model has only little explicit semantics, the conceptualization of Newman is already a well-defined ontology implemented in the *OWL Web Ontology Language*.

Newman's early ontology was followed by a number of other ontologies in the subsequent years. Table 9.1 lists nine ontologies we found in an extensive survey of the literature and web. It gives the main purpose of the ontologies along with the OWL sublanguage and OWL 2 profile they are compliant with. A detailed discussion of the ontologies can be found in Lohmann et al. (2011). An earlier review of a part of the ontologies is provided by Kim et al. (2008).

Table 9.1 Related ontologies in the domain of tagging and folksonomies that our semantic model is based on

Name	Authors	Release date (latest update)	Main purpose	OWL sublanguage	OWL 2 profile
Tag ontology	Newman et al.	2005-03-23 (2005-12-21)	First tagging ontology	OWL Full	–
Tagging ontology	Kner	2006 (2007-01-15)	Domain description	OWL Full	–
Ontology of folksonomy	Echarte et al.	2007 (–)	Domain description	OWL DL	–
Social semantic cloud of tags	Kim et al.	2007-03-23 (2008-06-13)	Tag clouds	OWL Full	–
Meaning of a tag	Passant & Laublet	2008-01-15 (–)	Semantic tagging	OWL Full	–
Upper tag ontology	Ding et al.	2008 (–)	Upper ontology	OWL Lite	OWL 2 RL
Common tag	Tori et al.	2009-06-08 (–)	Minimal ontology	OWL Full	–
TAGora tagging ontology	Szomszor et al.	2009 (2010)	Automatic disambiguation	OWL Lite	OWL 2 RL
NiceTag ontology	Limpens et al.	2009-01-09 (2010-09-09)	Taggings as speech acts	OWL Full	–

Though the ontologies describe many important aspects of tagging, none of them defines all of the aforementioned concepts and relationships needed for a comprehensive domain description. Taking one ontology and extending it is difficult due to various conceptual limitations. An integration and alignment of a part of the ontologies results in similar problems (Lohmann et al. 2011).

Hence, we finally decided to develop a new ontology that takes the best parts of the reviewed ontologies, adds missing pieces and combines all in one consistent conceptualization. To keep the ontology compact and understandable, we decided for a modular approach that separates rare and very specific concepts from the core ontology. Such a separation of concerns is well-known from ontologies and schemata like `SIOC` (with its `access`, `types`, and `services` modules) or `RSS` (with its `dc`, `syndication`, and `content` modules).[1] It also helps to keep the core ontology relatively stable with regard to future changes in the domain of tagging and folksonomies.

Certain concepts of the domain, such as the resources that are being tagged or the users who perform the tagging, are already well described in other contexts or more general ontologies. Therefore, we also surveyed ontologies of related domains.

[1] Ontologies and schemata are abbreviated by their common namespace prefixes in the following. The namespace prefixes and URIs of all referenced vocabularies are given in Table 9.2 at the end of this chapter.

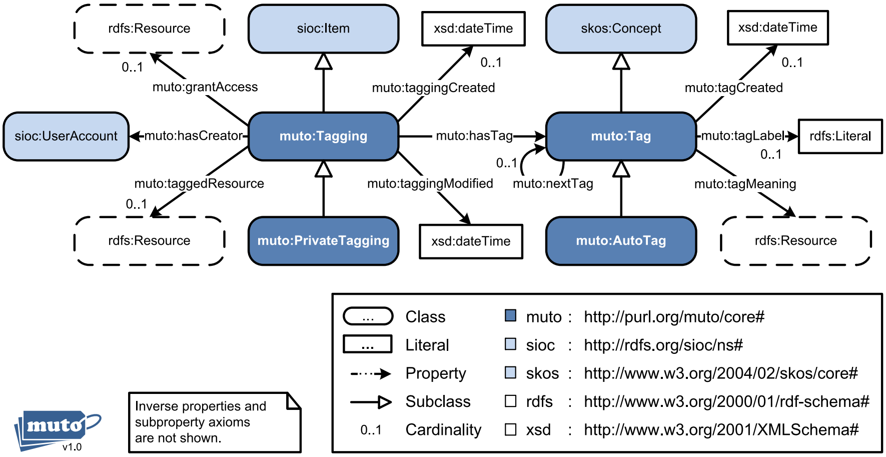


Fig. 9.1 Core concepts and relationships of the Modular Unified Tagging Ontology (MUTO)

In particular, we considered ontologies that are widely used and investigated, such as SKOS or FOAF, as we can expect their conceptualizations to be comparatively mature and stable.

9.4 An Ontology for Tagging and Folksonomies

Based on the previous considerations, we developed the *Modular Unified Tagging Ontology (MUTO)* to formally describe the semantic model. Like the related tagging ontologies listed in Table 9.1, it is implemented in the OWL Web Ontology Language. This language is based on the *Resource Description Framework (RDF)* and closely related to *RDF Schema (RDFS)*. All three languages are recommendations of the *World Wide Web Consortium (W3C)*. With regard to the first version of OWL, MUTO is compliant to the sublanguage OWL Lite; with regard to the second OWL version, it is compliant with the OWL 2 RL profile. This profile fits particularly well in our case, as it is recommended for "relatively lightweight ontologies [that] are used to organize large numbers of individuals" and approaches "where it is useful or necessary to operate directly on data in the form of RDF triples" (W3C OWL Working Group 2012).

We presented an earlier version of MUTO in (Lohmann et al. 2011). In the following, we describe the core conceptualization of version 1.0 of the ontology, as it is depicted in Fig. 9.1. Note that inverse properties and subproperty axioms are not shown in this compact diagram. The complete specification is publicly available on the web at the persistent URL http://purl.org/muto.

The ontology defines two core classes, one for the taggings A (muto:Tagging) and one for the tags T (muto:Tag), which form the center of the ontology. They are both specializations of more general classes from the well-known SIOC and

SKOS vocabularies. The other two key concepts, the resources R and users U, are not unique to tagging. We do not need to define new classes or specializations here, as we can directly reuse concepts from existing vocabularies, namely `sioc:UserAccount` and `rdfs:Resource`. Based on these four key concepts, we present the MUTO ontology in more detail in the following.

9.4.1 Taggings

The central `muto:Tagging` class describes the taggings, i.e. the set of annotations A. It contains the n-ary relations that link the resources, tags, and users. Using classes to represent n-ary relations is well-known from many modeling languages (e.g., UML with its *association class* construct) and common practice in OWL (Noy et al. 2006).

`muto:Tagging` is defined as a subclass of `sioc:Item`. We regard this as an adequate alignment, since SIOC has been designed to describe "user-generated content" from "online community sites" (Bojars et al. 2008). Apart from `sioc:Item`, there is a number of other concepts of the SIOC vocabulary that can be fruitfully reused in the domain of tagging. For instance, we do not need to create a new concept for comments assigned to taggings, as we can take `sioc:note`. Likewise, we can reuse concepts of SIOC to describe the source of tagging by first grouping taggings with `sioc:Container` and then linking them to a joined source with `sioc:has_space`.

9.4.2 Tags

The second core class of the MUTO ontology is `muto:Tag`. It describes the set of tags T. Each tag is an instance of this class with its own URI. Using class instances for tags instead of simple literals allows for the definition of tag properties, such as the later described `muto:tagMeaning` and `muto:nextTag`.

It is important to note that tags with the same label are not merged in MUTO, as this would not only affect the labels but also other tag properties. In our understanding, aggregations of tags with the same label (e.g. for the generation of *tag cloud* visualizations) are not a part of the semantic model but are rather performed by the interactive system. However, the MUTO core ontology can be extended by a module for aggregated tags if this information should be included in the semantic model.

Semantically, tags are very close to what is commonly represented by `skos:Concept`. We thus made `muto:Tag` a subclass of `skos:Concept`, which results in similar benefits to those described above for the subclassing of `sioc:Item`. For instance, it allows us to reuse SKOS concepts in MUTO, such as `skos:narrower` and `skos:broader` to represent hierarchical relations between tags. Likewise, `skos:related` can be used to describe tag relations of a more general nature. The description of other tag relations is not part of the MUTO core ontology, but could easily be integrated with a corresponding module if needed.

The only tag relation we explicitly defined in the MUTO core ontology is `muto:nextTag` (and its inverse counterpart `muto:previousTag`) to describe the sequential order in which tags are entered by the users during the act of tagging. Usually, people enter more than one tag per tagging (Halpin et al. 2007) and they expect the ordering of the tags to remain the same whenever they access a tagging. Using property relations to represent sequences is common practice in OWL (Drummond et al. 2006).

MUTO strictly distinguishes between tags (which have exactly one label according to Axiom 3) and concepts (which can have more than one label). However, it supports the mapping between tags and concepts with the property `muto:tagMeaning`. This is particularly useful in the aforementioned case of *semantic tagging* where the meaning of tags is made explicit by linking them to well-defined entities, such as DBpedia resources (see Sect. 9.2.2 and the example in Sect. 9.5). `muto:tagMeaning` can also be used to indicate synonym tags, simply by linking all tags with identical meaning to the same resource. This includes different tags that are variations of the same term (e.g. if one tag has an underscore as delimiter and the other a hyphen).

9.4.3 Users

MUTO reuses `sioc:UserAccount` to represent the accounts of the users who created the taggings (i.e. the set of users U). Linking users by their accounts is more accurate and flexible than linking them directly (e.g. by using `foaf:Person`), as it allows one user to have several accounts (e.g. one for work-related and one for personal taggings).

An alternative to `sioc:UserAccount` would have been the semantically closely related class `foaf:OnlineAccount`. We decided for the `SIOC` variant because we also used other concepts of this vocabulary along with `muto:Tagging` and can thus stay in one namespace. Moreover, it provides flexible support for group tagging (see Sect. 9.2.2) with its class `sioc:Usergroup` that `sioc:UserAccount` can be linked to. Yet, since `sioc:UserAccount` is a subclass of `foaf:OnlineAccount`, concepts from the `FOAF` vocabulary can also be used to describe users and user-related information.

9.4.4 Resources

Resources are linked to taggings by the property `muto:taggedResource`. Like `muto:tagMeaning` and `muto:grantAccess`, the property has no explicit range and can thus be linked to all instances of `rdfs:Resource`, as indicated in Fig. 9.1. Since `rdfs:Resource` is “the class of everything” and “all other classes are subclasses of this class” in RDF (Brickley and Guha 2004), it means that taggings can be linked to any kind of resource. This is in line with the general idea of tagging, where the interactive system determines what is considered a resource.

9.4.5 Further Concepts

In addition, we reused concepts from the Dublin Core vocabulary (DCMI Usage Board 2012) to enrich the instances of both `muto:Tagging` and `muto:Tag` with date and time information. Instead of directly linking Dublin Core properties, we defined own subproperties in order to equip them with exact domain and range axioms. We defined two such properties for `muto:Tagging`, one describing the creation date and time (`muto:taggingCreated`) and the other one tracking every single edit (`muto:taggingModified`). The latter can be useful, for instance, to sort taggings by date of last modification.

The date and time information of tags (`muto:tagCreated`) is conceptually separated from that of taggings. This is useful if certain tags of a tagging are added at a later time. Omitting the separate date and time information in these cases can result in biased tag statistics and wrong conclusions about the evolution of the folksonomy. However, since the creation dates and times of tags are usually the same as the creation dates and times of the associated taggings, we defined `muto:tagCreated` as an optional property to prevent a redundant representation of this information. If no separate date and time information is given for a tag, it is assumed that the tag has been created at the same date and time as the associated tagging (i.e., `muto:tagCreated` = `muto:taggingCreated`). The ranges of all three subproperties `muto:tagCreated`, `muto:taggingCreated`, and `muto:taggingModified` are given with `xsd:dateTime` in order to force a standardized format and improve interoperability.

Finally, the MUTO core ontology includes the domain-specific concepts of *private* and *automatic* tagging (see Sect. 9.2.2). They are defined as specializations of the central `muto:Tagging` and `muto:Tag` classes. A private tagging is only visible to its creator, unless the creator has not explicitly granted access to it by others, as expressed by the `muto:grantAccess` property. Note that the ontology can only provide a description of the concept; the correct implementation of privacy constraints remains the duty of the interactive system.

As automatic tagging denotes tagging with automatic tags, `muto:AutoTag` is a subclass of `muto:Tag`. Describing manual and automatic tagging in the same ontology makes sense, as it avoids redundant modeling and allows for an easier transformation of automatic tags into manual (i.e. user validated) ones. In addition, it allows to associate both types of tags with the same tagging instance, which is conceptually consistent with how automatic tags are often applied, namely as a complement to the tags entered by the user.

Note that there is no need to define the 'counterpart' concepts of public and manual tagging, as these are the default modes, i.e. the usual case is public tagging with user-assigned tags. Therefore, taggings that are not instances of `muto:PrivateTagging` are public by default. Likewise, all tags that are not instances of `muto:AutoTag` are assumed to be manually entered.

9.4.6 Cardinality Constraints

We equipped the MUTO ontology with cardinality constraints to ensure that the fundamental principles of tagging (as defined by the axioms in Sect. 9.2.1) are not violated. Since these fundamental principles are important for the interoperability and processability of folksonomies, we decided to define them globally. That is, we used functional properties (`owl:FunctionalProperty`) instead of property restrictions (`owl:Restriction`), what is different from earlier versions of MUTO which used property restrictions exclusively (Lohmann et al. 2011). This decision was also motivated by the fact that the maximum cardinality is one for all properties that need to be constrained in MUTO. Furthermore, functional properties do not force their use but can be omitted, keeping MUTO flexible despite all constraints.

Taggings and tags are linked by the property `muto:hasTag`. The cardinality of this property is not restricted, as there is no restriction on the number of tags that a tagging may consist of (see Axiom 1). MUTO even allows for taggings without tags to support cases where users first simply index a resource and add tags later, as possible in some systems (e.g. Delicious).

In contrast, we defined `muto:taggedResource` to be functional, since each tagging is uniquely linked to a single resource according to Axiom 1. The axiom states the same for the user account, i.e. each tagging is created by one user. Hence, `muto:hasCreator` is also a functional property in MUTO.

As tags have always exactly one label (Axiom 3), `muto:tagLabel` has also a cardinality of one. Finally, `muto:taggingCreated`, `muto:tagCreated`, and `muto:nextTag` have a cardinality of one, since multiple definitions of these properties for the same instance would not make sense. Apart from these cardinality constraints we have avoided to overly specify the ontology. In particular, we did not use OWL constructs that are not part of OWL Lite, such as `owl:disjointWith` or `owl:unionOf`, to not unnecessarily increase the formal complexity of the ontology.

9.5 Application Example

Figure 9.2 depicts a sample scenario of using the MUTO ontology with the social bookmarking system Example.org. It shows the RDF graph of user Alice who an-

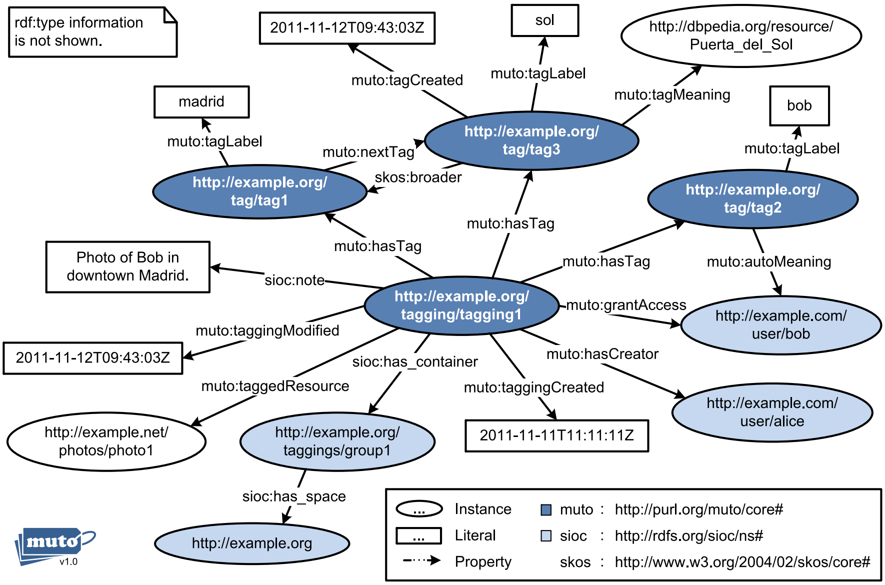

rdf:type information is not shown.
2011-11-12T09:43:03Z
sol
http://dbpedia.org/resource/Puerta_del_Sol
muto:tagCreated
muto:tagLabel
muto:tagMeaning
madrid
bob
muto:tagLabel
muto:nextTag
http://example.org/tag/tag3
muto:tagLabel
http://example.org/tag/tag1
skos:broader
http://example.org/tag/tag2
muto:hasTag
muto:hasTag
Photo of Bob in downtown Madrid.
muto:hasTag
muto:autoMeaning
sioc:note
http://example.org/tagging/tagging1
muto:grantAccess
http://example.com/user/bob
muto:taggingModified
2011-11-12T09:43:03Z
sioc:has_container
muto:hasCreator
muto:taggedResource
muto:taggingCreated
http://example.com/user/alice
http://example.net/photos/photo1
http://example.org/taggings/group1
2011-11-11T11:11:11Z
sioc:has_space
muto v1.0
http://example.org
Instance
Literal
Property
muto : http://purl.org/muto/core#
sioc : http://rdfs.org/sioc/ns#
skos : http://www.w3.org/2004/02/skos/core#

Fig. 9.2 Example RDF graph depicting a private tagging of a photo by user Alice

notated a photo from the website Example.net. Assume that Alice interacts with a well-designed user interface. She does not get in touch with the ontology but it is rather used for the design and internal representation in the interactive system and/or for sharing the folksonomy with other systems. Listing 9.1 provides the OWL code of the example in RDF/Turtle format.

9.5.1 Scenario

Imagine the following scenario that led to the creation of the tagging instance:[2] Alice logs into her account of the social networking service Example.com (`sioc:hasCreator`). From there, she uses the social bookmarking system Example.org to annotate a photo she uploaded to the media sharing website Example.net (`muto:taggedResource`). As the photo shows her friend Bob in downtown Madrid, she starts tagging with entering the tag 'madrid' (`muto:Tag`). Then, she recognizes that the system has automatically identified Bob on the photo and added his name as a tag (`muto:AutoTag`). The system got his name (and further information) from the social networking service Example.com, of which Bob is also a member. In addition, the system links Bob's name to his account (`muto:autoMeaning`, a subproperty of `muto:tagMeaning`). Though Alice

[2] In brackets, we give the ontology classes and properties used to represent the information.

```
@prefix muto: <http://purl.org/muto/core#> .
@prefix sioc: <http://rdfs.org/sioc/ns#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
<http://example.org/tagging/tagging1> a muto:PrivateTagging;
    muto:taggedResource <http://example.net/photos/photo1>;
    muto:hasCreator <http://example.com/user/alice>;
    muto:hasTag <http://example.org/tag/tag1>,
        <http://example.org/tag/tag2>,
        <http://example.org/tag/tag3>;
    muto:taggingCreated "2011-11-11T11:11:11Z";
    muto:taggingModified "2011-11-12T09:43:03Z";
    muto:grantAccess <http://example.com/user/bob>;
    sioc:has_container <http://example.org/taggings/group1>;
    sioc:note "Photo of Bob in downtown Madrid." .
<http://example.org/tag/tag1> a muto:Tag;
    muto:tagLabel "madrid" ;
    muto:nextTag <http://example.org/tag/tag2>.
<http://example.org/tag/tag2> a muto:Tag;
    muto:tagLabel "sol";
    muto:tagCreated "2011-11-12T09:43:03Z";
    muto:tagMeaning <http://dbpedia.org/resource/
        Puerta_del_Sol>;
    skos:broader <http://example.org/tag/tag1> .
<http://example.org/tag/tag3> a muto:AutoTag;
    muto:tagLabel "bob";
    muto:autoMeaning <http://example.com/user/bob> .
<http://example.org/taggings/group1> a sioc:Container;
    sioc:has_space <http://example.org> .
```

Listing 9.1 OWL code of the example in RDF/Turtle format ('a' = rdf:type)

marked the tagging as private (`muto:PrivateTagging`), she decides to share it with Bob and grants him access (`muto:grantAccess`). She also adds a comment to the tagging describing the contents of the photo (`sioc:note`).

One day later, Alice looks at the photo again and recognizes that it was taken at Puerta del Sol, a central square in Madrid. She opens the tagging and adds the tag 'sol' (`muto:Tag`) to the previously assigned tags 'madrid' and 'bob'. Furthermore, she decides to 'semantify' the tag so that she will later remember what 'sol' means. First, she makes 'sol' a subtag of 'madrid' (`skos:broader`) to indicate that it is a specific location in Madrid. Second, she gives the tag explicit meaning by linking it to the corresponding DBpedia resource (`muto:tagMeaning`).

The information that the tag 'sol' was added at a later time than the other two tags is given by its property `muto:tagCreated`. This property would not be necessary if the tag would have been entered along with the others. Accordingly, the timestamp of `muto:tagCreated` is the same as of `muto:taggingModified` but different from the one of `muto:taggingCreated`. Source information is linked with `sioc:has_space` after the tagging has been assigned to a container with `sioc:has_container`.

9.5.2 Discussion

The example illustrates an advanced case of tagging that uses many of the concepts from the semantic model. The most basic variant of tagging—a list of tags without disambiguations, hierarchical relations, comments, or automatic tags—can be described with much fewer concepts. This capability of supporting different levels of tagging, from simple to semantic, from manual to automatic, and from public to private, was one of the main goals in the development of the semantic model. On the other hand, we avoided to make the semantic model unnecessary complex but kept it understandable to people who use it. Finding a good balance between comprehensiveness and compactness was thus another major goal in the development of the semantic model.

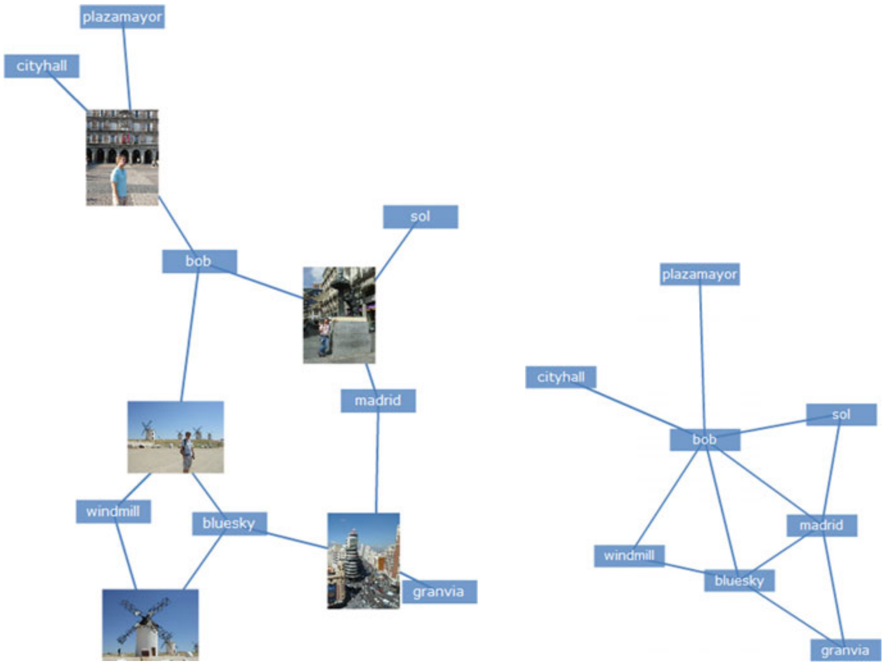
The example also indicates the benefits of a precise domain description for the development of interactive systems. There would be many different ways to represent the information from the scenario; having one common conceptualization helps to create a joint understanding among the developers of an interactive system. More importantly, the semantic model can also increase the interoperability between different interactive systems, as illustrated by the scenario: It links taggings of a social bookmarking system with photos from a media sharing website and user profiles of a social networking service.

9.6 Visualization

If folksonomies are based on the semantic model, they become independent from individual interactive systems. This opens up possibilities for developing user interface components that can be used in multiple systems. In the following, we illustrate these possibilities on the example of graph visualizations. Such visualizations nicely depict the core structure of folksonomies, as described by the annotations A which link the resources R, tags T, and users U. If we consider the MUTO ontology, these sets are represented by the classes `rdfs:Resource`, `muto:Tag`, `sioc:UserAccount`, and `muto:Tagging`.

This core structure forms a hypergraph that can be split into several subgraphs representing specific parts of the folksonomy (Mika 2005). In particular, we derive three bipartite graphs $G(RT)$, $G(TU)$, and $G(UR)$ that describe the relations between each two of the basic element sets.

An example for the graph $G(RT)$ with $V = R \cup T$ is shown in Fig. 9.3a. The nodes represent the resources and tags, i.e. instances of `rdfs:Resource`



(a) (b)

Fig. 9.3 Graph visualizations: (**a**) photos linked by tags, (**b**) links between tags without photos

(or some more specific subclass) and `muto:Tag`, while the edges represent the links between them. These links are given by the `muto:hasTag` and `muto:taggedResource` property relations. Unlike in the semantic model, we merged tags with same labels in the graph visualization.

Continuing from the above scenario, assume that the graph was generated from the taggings of user Alice. The resources are a selection of photos which Alice uploaded to the media sharing website Example.net and which she annotated with tags. The graph visualization helps to explore relationships between the photos. We see, for instance, that there are other photos also showing Bob. These photos can then be selected and viewed in detail.

All three bipartite graphs $G(RT)$, $G(TU)$, and $G(UR)$ derived from the basic folksonomy structure can be further reduced, depending on what the user is interested in. If we split $G(RT)$ again, we get two graphs showing the set of resources and the set of tags, respectively. An example of the latter is shown in Fig. 9.3b. It visualizes the same tags as graph $G(RT)$ from Fig. 9.3a but without the resources, i.e. the tagged photos. Instead, it shows only the links that result from the tagging of the photos. This kind of visualization is sometimes called *tag graph*, as it is related to tag cloud visualizations but additionally displays the links between the tags based on their co-use (Lohmann and Díaz 2012). It can be extended in various directions, e.g. by weighting the font sizes of the tags like in tag clouds or by weighting the tag relations.

Apart from that, a large number of other graph visualizations can be derived from folksonomies; in particular, if we consider not only the basic folksonomy tuple $F = (R, T, U, A)$ but also other concepts of the semantic model, such as time information (`muto:taggingCreated`). Generally speaking, we are free to create all kinds of user interfaces, as long as they are consistent with the semantic model. This allows even for adaptable user interface components where users select the parts of the folksonomy they are interested in (e.g. choose from a selection of different graph visualizations). Thus, not only the folksonomies become independent from individual interactive system but also the related user interface components.

9.7 Conclusions and Future Work

In this chapter, we have presented a semantic model for the domain of tagging and folksonomies. It not only contributes to a better domain understanding but also improves the interoperability of folksonomies and the reusability of related user interface components. It is formally described in an OWL ontology allowing for machine interpretation and automated reasoning. We have illustrated how the model can support the graph-based representation and visualization of folksonomies. If folksonomies are based on it, they can be shared among interactive systems and the same program code (e.g. SPARQL queries) can be used to access and visualize parts of the folksonomies. This allows for the development of reusable and adaptable user interface components which offer different perspectives on folksonomies depending on the interests of the users.

Our goal was to develop a compact yet comprehensive semantic model that considers all important concepts and relationships in the domain of tagging and folksonomies. Furthermore, we wanted to keep it as understandable as possible for the people who use it. If we compare it with existing conceptualizations, it is most closely related to the "Tag Ontology" developed by Newman (2005) and the "Bookmark Schema" of the Annotea project (Koivunen 2006). However, it additionally considers several concepts that are missing in these approaches, such as some advanced tagging concepts described in the ontologies of Echarte et al. (2007) as well as Passant and Laublet (2008).

Two major challenges in the application of semantic models are performance and scalability. In this work, we have focused on a precise conceptualization rather than on a technical optimization for large folksonomies. In such cases, other representations that allow for a fast processing and efficient storage of the folksonomy might be more useful. Furthermore, specific modules may be integrated into the MUTO ontology that speed up processing, such as properties that directly link the resources, tags, and users in order to avoid the indirection via taggings. However, such pragmatic extensions should be used with care as they may lead to conceptual inconsistencies (e.g. direct relations between tags and resources can conflict with the concept of private tagging).

Table 9.2 Alphabetical list of the names, namespace prefixes, and URIs of the referenced vocabularies

Vocabulary name	Prefix	URI reference
Common tag	CTAG	http://commontag.org/ns#
Friend of a friend	FOAF	http://xmlns.com/foaf/0.1/
Meaning of a tag	MOAT	http://moat-project.org/ns#
Modular unified tagging ontology	MUTO	http://purl.org/muto/core#
NiceTag ontology	NT	http://ns.inria.fr/nicetag/2010/09/09/voc
Ontology of folksonomy	OF	http://www.eslomas.com/tagontology-1.owl
RDF schema	RDFS	http://www.w3.org/2000/01/rdf-schema#
RDF site summary	RSS	http://purl.org/rss/1.0/
Semantically-interlinked online communities	SIOC	http://rdfs.org/sioc/ns#
Simple knowledge organization system	SKOS	http://www.w3.org/2004/02/skos/core
Social semantic cloud of tags	SCOT	http://scot-project.org/scot/ns#
Tag ontology	TAGS	http://www.holygoat.co.uk/owl/redwood/0.1/tags/
Tagging ontology	TO	http://bubb.ghb.fh-furtwangen.de/TagOnt/tagont.owl
TAGora tagging ontology	TT	http://tagora.ecs.soton.ac.uk/schemas/tagging
Upper tag ontology	UTO	http://info.slis.indiana.edu/dingying/uto.owl
XML schema	XSD	http://www.w3.org/2001/XMLSchema#

Future work includes the application and extension of the semantic model in various contexts. In particular, we are interested in ontology modules that add advanced tagging concepts to represent, for instance, specific types of tags (geotags, star ratings, etc.) or tag relations (synonymy, part-of, etc.). Furthermore, we plan to test the semantic model with different interactive systems to evaluate its general applicability and identify issues for improvement.

References

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., et al. (2009). DBpedia—a crystallization point for the web of data. *Journal of Web Semantics*, *7*(3), 154–165.

Bojars, U., Breslin, J. G., Peristeras, V., Tummarello, G., & Decker, S. (2008). Interlinking the social web with semantics. *IEEE Intelligent Systems*, *23*(3), 29–40.

Brickley, D., & Guha, R. V. (2004). *RDF vocabulary description language 1.0: RDF schema*. http://www.w3.org/TR/rdf-schema/.

DCMI Usage Board (2012). *DCMI metadata terms*. http://dublincore.org/documents/dcmi-terms/.

Drummond, N., Rector, A., Stevens, R., Moulton, G., Horridge, M., Wang, H., et al. (2006). Putting OWL in order: patterns for sequences in OWL. In *CEUR-WS.org: Vol. 216. Proceedings of the OWLED '06 workshop on OWL: experiences and directions*.

Echarte, F., Astrain, J. J., Córdoba, A., & Villadangos, J. E. (2007). Ontology of folksonomy: a new modelling method. In *CEUR-WS.org: Vol. 289. Proceedings of the semantic authoring, annotation and knowledge markup workshop (SAAKM '07)*.

Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, *5*(2), 199–220.

Gruber, T. (2005). *TagOntology—a way to agree on the semantics of tagging data*. http://tomgruber.org/writing/tagontology.htm.

Gruber, T. (2007). Ontology of folksonomy: a mash-up of apples and oranges. In *Proceedings of the 1st conference on metadata and semantics research (MTSR '05). International Journal on Semantic Web and Information Systems*, *3*(2), 1–11.

Halpin, H., Robu, V., & Shepherd, H. (2007). The complex dynamics of collaborative tagging. In *Proceedings of the 16th international conference on world wide web (WWW '07)* (pp. 211–220). New York: ACM.

Heymann, P., & Garcia-Molina, H. (2006). *Collaborative creation of communal hierarchical taxonomies in social tagging systems* (Technical Report No. 2006-10). Stanford InfoLab.

Hotho, A., Jäschke, R., Schmitz, C., & Stumme, G. (2006). Information retrieval in folksonomies: search and ranking. In *LNCS: Vol. 4011. Proceedings of the 3rd European semantic web conference (ESWC '06)* (pp. 411–426). Berlin: Springer.

Kim, H. L., Scerri, S., Breslin, J. G., Decker, S., & Kim, H. G. (2008). The state of the art in tag ontologies: a semantic model for tagging and folksonomies. In *Proceedings of the international conference on Dublin core and metadata applications (DC '08)* (pp. 128–137). Dublin: DCMI.

Koivunen, M.-R. (2006). Semantic authoring by tagging with annotea social bookmarks and topics. In *CEUR-WS.org: Vol. 209. Proceedings of the 1st semantic authoring and annotation workshop (SAAW '06)*.

Li, R., Bao, S., Yu, Y., Fei, B., & Su, Z. (2007). Towards effective browsing of large scale social annotations. In *Proceedings of the 16th international conference on world wide web (WWW '07)* (pp. 943–952). New York: ACM.

Lohmann, S., & Díaz, P. (2012). Representing and visualizing folksonomies as graphs: a reference model. In *Proceedings of the international working conference on advanced visual interfaces (AVI '12)* (pp. 729–732). New York: ACM.

Lohmann, S., Díaz, P., & Aedo, I. (2011). MUTO: the modular unified tagging ontology. In *Proceedings of the 7th international conference on semantic systems (I-SEMANTICS '11)* (pp. 95–104). New York: ACM.

Mika, P. (2005). Ontologies are us: a unified model of social networks and semantics. In *LNCS: Vol. 3729. Proceedings of the 4th international semantic web conference 2005 (ISWC '05)* (pp. 522–536). Berlin: Springer.

Miller, G. A. (1995). Wordnet: a lexical database for English. *Communications of the ACM*, *38*, 39–41.

Newman, R. (2005). *Tag ontology design*. http://www.holygoat.co.uk/projects/tags.

Noy, N., & McGuinness, D. (2001). *Ontology development 101: a guide to creating your first ontology* (Technical Report No. SMI-2001-0880). Stanford Medical Informatics.

Noy, N., Rector, A., Hayes, P., & Welty, C. (2006). *Defining N-ary relations on the semantic web*. http://www.w3.org/TR/swbp-n-aryRelations/.

Passant, A., & Laublet, P. (2008). Meaning of a tag: a collaborative approach to bridge the gap between tagging and linked data. In *CEUR-WS.org: Vol. 369. Proceedings of the WWW2008 workshop on linked data on the web (LDOW '08)*.

Peters, I. (2009). *Folksonomies: indexing and retrieval in web 2.0*. Berlin: de Gruyter.

Smith, G. (2008). *Tagging: people-powered metadata for the social web*. Berkeley: New Riders.

W3C OWL Working Group (2012). *OWL 2 web ontology language document overview* (2nd ed.). http://www.w3.org/TR/owl2-overview/.

Wu, X., Zhang, L., & Yu, Y. (2006). Exploring social annotations for the semantic web. In *Proceedings of the 15th international conference on world wide web (WWW '06)* (pp. 417–426). New York: ACM.