Semantic Web – Interoperability, Usability, Applicability 0 (2011) 1–0
IOS Press

A Classification of Semantic Annotation Systems

Editor(s): Krzysztof Janowicz, University of California, Santa Barbara, USA
Solicited review(s): Jérôme Euzenat, INRIA Grenoble Rhône-Alpes, France and Sebastian Tramp, Universität Leipzig, Germany

Pierre Andrews [a], Ilya Zaihrayeu [a] , and Juan Pane [a]
[a] *Dipartimento di Ingegneria e Scienza dell'Informazione*
The University of Trento, Italy
E-mail: {andrews,ilya,pane}@disi.unitn.it

Abstract.
The *Subject-Predicate-Object* triple annotation system is now well adopted in the research community, however, it does not always speak to end-users. In fact, explaining all the complexity of semantic annotation systems to laymen can sometime be difficult. We believe that this communication can be simplified by providing a meaningful abstraction of the state of the art in semantic annotation models and thus, in this article, we describe the issue of semantic annotation and review a number of research and end-user tools in the field. Doing so, we provide a clear classification scheme of the features of annotation systems. We then show how this scheme can be used to clarify requirements of end-user use cases and thus simplify the communication between semantic annotation experts and the actual users of this technology.

Keywords: Semantic, Semantic Annotation, Vocabulary, Classification Scheme, Tag, Attributes, Ontology

1. Introduction

Social annotation systems such as Delicious[1], Flickr[2] and others have laid the fundamentals of the Web 2.0 principles and gained tremendous popularity among Web users. One of the factors of success for these systems is the simplicity of the underlying model, which consists of a resource (e.g., a web page), a tag (normally, a text string), and a user who annotated the resource with the tag as can be seen in Figure 1. Despite its simplicity, the annotation model enables a set of useful services for the end user, e.g., searching resources using tags added by a community of users, computing the most popular tags and building the so-called tag clouds[3], finding users with common interests based on the resources they annotated and on the tags they used and providing a recommendation service on this basis, among others.

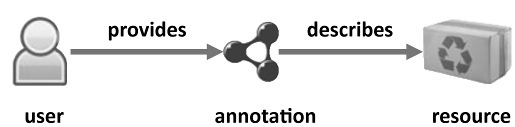

Fig. 1. A Generic Annotation Model

Due to the natural language nature of the underlying model, these systems have been criticised for not being able to take into account the explicit information about the meaning or semantics of each tag. For example, different users can use the same tag with different meanings (i.e., homonyms), different tags with the same meaning (i.e., synonyms), different tags in the same meaning but at different levels of abstraction, morphological variations of the same tag, and so on. In addition to the problem of ambiguity in the interpretation of the meaning of the tag itself, there is a further problem of deciding what the tag refers to. For exam-

[1] http://www.delicious.com/
[2] http://www.flickr.com/
[3] http://en.wikipedia.org/wiki/Tag_cloud

ple, a tag "John" attached to a photo does not specify if John is a person on the photo or the photographer who took this photo.

With the advent of semantic technologies, some of the aforementioned problems were (partially) addressed in some systems. For example, Flickr introduced the so-called machine tags that use a special syntax to define extra information about the tag. For example, a tag "upcoming:event=428084" assigned to a photo encodes that it is related to an Upcoming.org event identified with "428084". Another example is Faviki[4] – a social bookmarking system similar to Delicious. However, when users enter a tag, they are asked to disambiguate its meaning to a known concept extracted from Wikipedia.

There are many types of annotation models available in the scientific state of the art and in the already existing end-user applications. In the semantic web community, these models have been abstracted by the *Subject-Predicate-Object* triple that can be used for most of the annotation types discussed here. However, this generalisation is low level in the sense that its atomic building element, a triple, can be used to represent a single property of an object, whereas end user applications can use structurally richer atomic building blocks and include objects such as people, events, and locations.

In Sections 2, 3 and 4, we present an extended survey of the tools and models that were available and we provide what we believe to be a simple abstraction of the important dimensions that an annotation model can have. The features were selected based on the intention to demonstrate how much they can contribute to the definition of a semantic annotation model. The features were grouped in the following three dimensions:

1. the structural complexity of annotations (e.g., tags, attributes, and relations), see Section 2;
2. the vocabulary type, i.e., the level of formality of annotations defined on the basis of the form of the underlying knowledge organisation system to the elements of which the annotations can be linked (e.g., thesauri, taxonomies), see Section 3; and
3. the user collaboration in sharing and reusing semantic annotations and in the collaborative construction and evolution of the underlying knowledge organisation system, see Section 4.

[4]http://faviki.com

Within each dimension we provide a specification of the most typical approaches by giving their description, comparative analysis of their advantages and disadvantages, and examples of popular systems (both in the research field and among publicly available systems). The goal of the comparative analysis is to show the trade off between the level of user involvement and provided services in each approach. The term user encompasses the notion of user as annotator and user as consumer of the annotations, wherever it requires clarification, we will use the term annotator or user as consumer to differentiate one from the other.

In Section 6 we show how this annotation model classification can be used to elicit requirements from end-users. To do this, we provide summaries and conclusions made from the analysis of a concrete use case pertaining to the telecommunications sector.

2. Structural Complexity of Annotations

The first dimension that we discuss is the structural complexity of annotations. This dimension relates to the amount of information that is encoded in the annotation itself, how it is structured in the underlying storage model and how this structure can be used. This structure has great influence on what data can be displayed to the user, how it can be displayed, but also what type of back-end services can be provided. We distinguish between *tags*, *attributes*, *relations* and *ontologies*. Tags are at the beginning of the spectrum and represent the easiest form of annotation from the annotator point of view; whereas ontologies are at the other end of the spectrum and represent the hardest form of annotation from the annotator point of view. In fact, as it has been showed in [1], designing ontologies is a difficult and error-prone task even for experienced users.

2.1. Tags

A tag annotation element is a non-hierarchical keyword or free-form term assigned to a resource (see Figure 2). A tag *implicitly* describes a particular property of a resource as the computer and other consumers of the annotation do not know the meaning that the annotator intended (except if the natural language used is unambiguous). Normally, a tag is a single word or a sequence of characters without spaces (which typically serve as tag separators in the user input). Examples of tags include: the name of the person on a picture, the name of the place where a picture was taken, or a topic

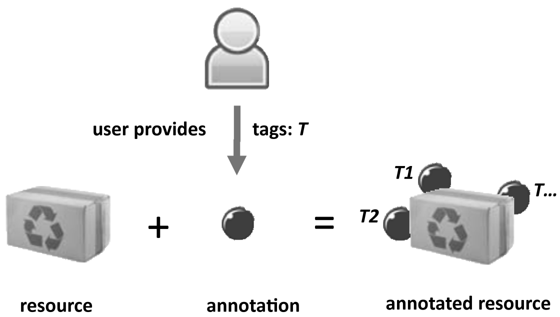

Fig. 2. The Tag Annotation Model

of a news article. The tagging annotation systems are often discussed as part of the folksonomy annotation model [2] that links tags, resources and users (annotators) following the model in Figure 1.

PROS With the increasing amount of applications applying Web 2.0 principles, the notion of tagging as a simple organisation system is now familiar for Internet users and, therefore, it entails nearly no learning curve for a typical user in order to start using them. They allow the user to easily annotate a Web resource with a free-text term and find other resources which were annotated with the same tag by browsing or searching. In fact, after four years of its existence Flickr reported to have "about 20 million unique tags" (January 2008) and 5 billion images (September 2010) [3].

CONS Tags represent a minimal annotation model from the structural complexity point of view and, therefore, can enable only a limited number of services mainly focused on basic retrieval and browsing (e.g., retrieve resources that were assigned tag `x`). Because they only implicitly describe resource properties, tags are subject to ambiguity in the interpretation of these properties. For example, natural language tag "John" attached to a picture does not specify whether John is a person on the picture or if he is the photographer who took the picture.

APPLICATIONS Delicious[5] is a social bookmaking service that allows its users to memorize and share URLs of Web resources such as blogs, articles, music, video, etc. It was founded in 2003 and now counts five million users and 150 million bookmarked URLs. The key idea of the service is that its users can access their bookmarks from any computer, for example at home, at work or while traveling. Delicious has been among the first popular systems that used tagging for organization and retrieval purposes. One of the important success factors of Delicious was its simplicity of use (see Figure 3). In a nutshell it works as follows: the user finds an interesting website and decides to bookmark it in Delicious; when adding the website the user annotates the website with a set of terms (called tags). Later, in order to retrieve a saved bookmark, the annotation consumer can query the system with one or more of the previously assigned tags. If the answer set contains too many elements, it can be refined by adding more terms to the query. Delicious allows the user to browse not only personal annotations, but also to find bookmarks saved and annotated by other annotators. Tagclouds (see Figure 4) and tag based search are starting points for navigation in the space of all (published) bookmarks of all users. While browsing bookmarks the system visualizes tags which were assigned to resources by other annotators.

Flickr[6] is a free image hosting service that stores over three billion images. It was launched in 2004 and popularized the concept of tagging together with Delicious. Flickr allows its users to upload their photos, organize and share them using tags. Even if, the users can establish relationships, form communities, comment and annotate photos of each other, the site is more used as a user's personal photo repository.

Another example of a popular social site which uses tags is Last.fm[7]. It is a UK-based Internet radio, founded in 2002, which has a thirty million active users in more than 200 countries. Last.fm allows its users to create custom playlists and radio stations from audio tracks available from the service's library. It also offers numerous social features such as recommendation of similar tracks considering user's favorites. Users can annotate resources such as bands, artists, albums and tracks and retrieve them using tag based search. Examples of other systems that use tags for annotating their resources include: Youtube[8], CiteULike[9] and LiveJournal[10].

Note that the *Subject-Predicate-Object* (*SPO*) model used widely in semantic web technologies (through RDF for instance) is already of higher structural complexity than the *tags* as it supposes the existence of a predicate linking the tag (Object) and the resource

[5] `http://www.delicious.com/`

[6] `http://www.flickr.org`
[7] `http://www.last.fm`
[8] `http://www.youtube.org`
[9] `http://www.citeulike.org/`
[10] `http://www.livejournal.com/`

Fig. 3. Annotating a Resource Using the Tag Annotation Model

Fig. 4. Flickr Cloud of Tags

(Subject). In fact, in the simple tagging annotation model, there is no need for this predicate as it is always meant to be the tagging relationship.

2.2. *Attributes*

An attribute annotation element is a pair $\langle AN, AV \rangle$, where AN is the name of the attribute and AV is the value of the attribute (see Figure 5). The attribute name defines the property of the annotated resource (e.g., “location”, “event”, “starting date”) and the attribute value specifies the corresponding value (e.g., “Trento”, “birthday”, “April 1, 2009”). Apart from this, the model allows us to define the data types for attributes and, therefore, enables type checking at query time.

Pros Attributes are pervasively used on the web and in desktop applications and, therefore, represent a well known notion for end-users. Differently from tags, attributes explicitly define the described resource properties and, therefore, enable a richer resource annotation and query language. For example, one could search for images of the Eiffel Tower taken between 1890 and 1900 by a specific photographer.

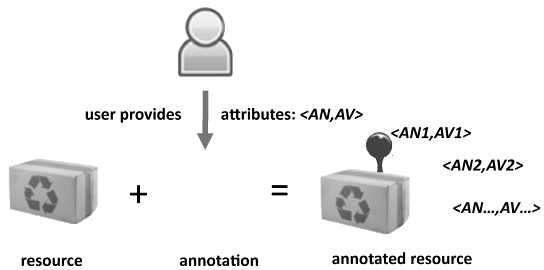

Fig. 5. The Attribute Annotation Model

Cons While enabling more services than tags do, attributes are still a limited means of annotation because they refer to single resources and, therefore, cannot be used to effectively enable services which are based on interrelationships that exist between resources (e.g., search and navigation between related resources). Furthermore, attribute annotations require a more metadata-knowledgeable annotator than tag annotations do.

Applications One of the earliest systems that used attributes for resource annotation was Semantic File Systems described by [4]. The system allows the user to assign arbitrary number of name-value pairs to the user's files and then retrieve them by creating so called virtual directories. The user creates virtual directories at runtime specifying the list of attributes. According to the user input the virtual directory contains only those files whose attributes match attributes from the list. The implementation of the ideas introduced in Semantic File systems can be found in search engines integrated directly in the operating systems. This is often referred to as Extended File Attributes[11] as files can be assigned a number of extra metadata attributes that are not relevant for the file system behaviour but can be used in search system such as Mac OS X Spotlight.

The Web has many examples of popular systems using attributes. Almost all social networks, (e.g. Facebook[12] and MySpace[13]) consider their users as a resource and use the attribute annotation model to represent user profiles. The variety of attributes is large and includes common attributes such as "Personal Info", "Contacts" as well as specific attributes, such as "Interests", "Traveling" (see Figure 6). Online markets such as Ebay[14] use the attribute annotation model to annotate resources which are items to be sold. The seller can assign an item with attributes such as "item location: Trento", "item price: $100", "shipping to: Italy", etc.

In the *Subject-Predicate-Object* model, the predicate would be used to link the resource (subject) and the attribute value (object) through a property name (predicate). In that sense, the RDF+OWL *Property* predicates can be considered to be a good example of the attribute complexity dimension.

To annotate cultural heritage resources in the Bibliopolis collection[15], [5] uses attributes representing metadata of scanned books, photos and related resources. They can then annotate visual resources with provenance, authors, date of publication, etc.

Both Delicious and Flickr tags systems were "hacked" by users to use so-called *machine tags* that allow to informally encode attributes in the simple tag system they provide. For instance, before the introduction of a specific geotagging interface by Flickr, users could geolocalise their photos by assigning tags of the form `geo:lat=XXXX` and `geo:long=YYYY`. Both websites have added adhoc support for searching for these machine tags.

[11] *xattr* in some unix based systems

[12] `http://www.facebook.com`

[13] `http://www.myspace.com/`

[14] `http://www.ebay.com`

[15] `http://www.bibliopolis.nl`

2.3. Relations

A relation annotation element is a pair $\langle Rel, Res \rangle$, where Rel is the name of the relation and Res is another resource. The relation name defines how the annotated resource is related with Res. At the conceptual level, the relation annotation model is an extension of the attribute annotation model to the domain of resources, which allows the user to interlink these resources (see Figure 7). For instance, in a scientific paper a citation referencing another paper is an example of a relation annotation which defines a relation between these documents.

PROS Relation annotations provide a way to interlink various resources through typed links. It allows the user to navigate from one resource to another and enable search and navigation based on these relation links.

CONS The annotator is expected to bear a higher mental load w.r.t. the previous models as, instead of describing one resource, the annotator has to understand what the two resources are about and what kind of relationship holds between them.

APPLICATIONS User of social network sites annotate their profiles establishing relationships between each other, which allow them to find friends and to meet new people by navigating the network of relations between users. Apart from this, some social networks, for instance Facebook, allow their users to annotate photos with links to profiles of people appearing in the picture (See Figure 8). In 2007, Facebook had around 1.7 billion uploaded photos with around 2.2 billion relation annotations. In a similar manner, bibliographic folksonomies such as Connotea [6] or BibSonomy[16] allow for relating bookmarked scientific references to their authors.

The relation annotation model can also be used to define relations within a resource. For example, the Araucaria project [7] annotates the rhetorical structure of a document using the RST relations annotation [8]. Figure 9 illustrates an example of intra document annotation of the RST relations; segments of text are given an identifier and links between each segment are annotated with a type of argumentation relationship (justify, condition, etc.).

Upcoming.org[17] is a social website for listing events that can be linked to Flickr photos by annotating them

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

[17] `http://upcoming.yahoo.com/`

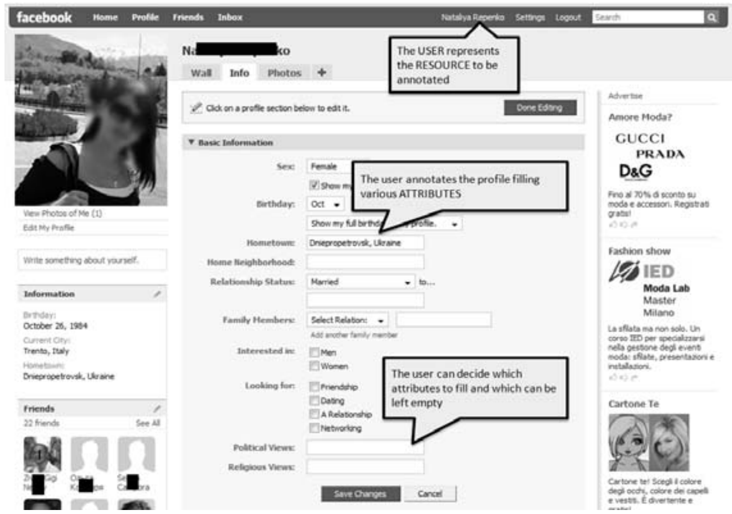

Fig. 6. Resource Annotation Using the Attribute Annotation Model

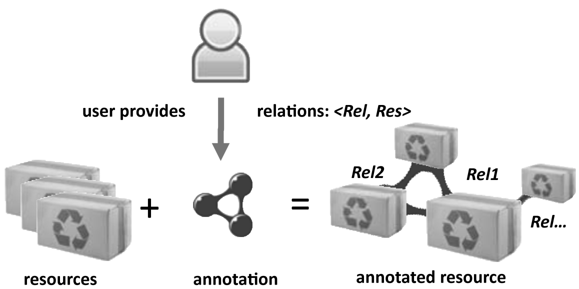

Fig. 7. The Relation Annotation Model

with so called triple or machine tag. Triple tags use a special syntax to define extra information about the tag making it machine readable. For example, a tag like “upcoming:event=428084” assigned to a photo encodes that it is related to Upcoming.org event identified with “428084”, therefore making these resources interlinked. Last.fm is another example of a system that use triple tags to link its tracks to Flickr photos.

Freebase[18] is a large knowledge base containing around five million various facts about the world. It is described as “an open shared database of the world’s knowledge” and “a massive, collaboratively-edited database of cross-linked data.”[19]. It allows its users to annotate resources (e.g. images, text, Web pages) using the Freebase annotation schema. The schema defines a resource annotation as a collection of attributes, its kind (e.g. person, event, location), and typed relations with other resources. It provides a convenient way to perform search and navigation in the space of resources allowing the user to find them using attributes, relations, and/or schema kinds.

2.4. Ontologies

This model is based on the notion of semantic annotation [9], a term coined at the beginning. It describes both the process and the resulting annotation or metadata consisting of aligning a resource or a part of it with a description of some of its properties and characteristics with respect to a formal conceptual model or ontology. Figure 10 shows an schematic representation of the model. As defined by Gruber et al., “an ontology is an explicit specification of a (shared) conceptualization” [10]. In practice, ontologies are usually modeled with (a subset of) the following elements: concepts (e.g., `CAR`, `PERSON`), instances of these concepts (e.g., `bmw-2333` *`is-instance-of`* `CAR`, `Marry` *`is-instance-of`* `Person`), properties of concepts and instances (e.g., `PERSON` *`has-father`*), restrictions on these properties (e.g., `MAX(PERSON` *`has-father`*`) = 1`[20]), relations between concepts (e.g., `PERSON` *`is-a`* `BEING`), relations between instances (e.g., `Marry` *`has-father`* `John`), etc [11]. The ontology annotation model allows the annotator to describe and interlink existing

[18] http://www.freebase.com

[19] http://www.crunchbase.com/company/metawebtechnologies

[20] The meaning of this rather informal notation is that any instance of the concept `PERSON` may have one father at most.

Fig. 8. Resource annotation using the relation annotation model

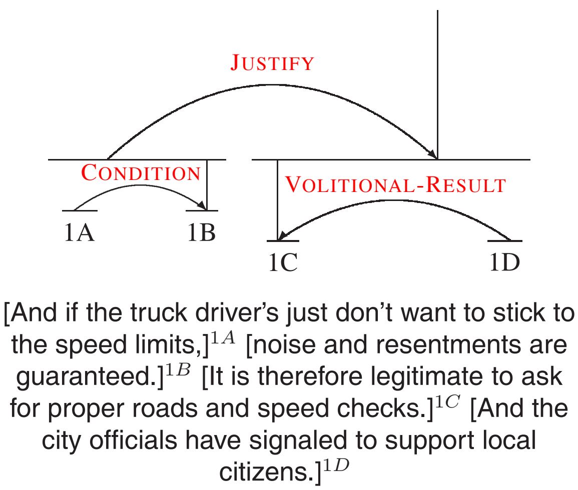

Fig. 9. Rethorical Structure Theory Relationship

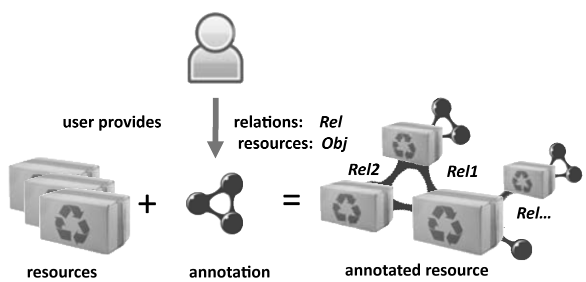

Fig. 10. The Ontology Annotation Model

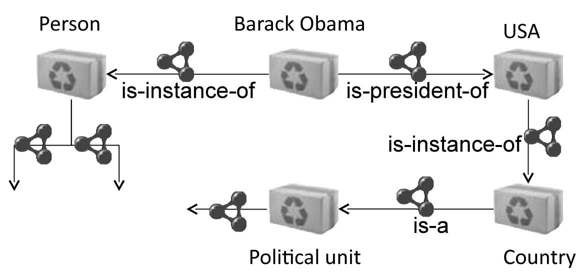

Fig. 11. An Example of the Application of the Ontology Annotation Model

resources by qualifying resources as concepts or as instances and by defining relations, properties, and restrictions that hold between them. Thus, it is the richest model from the structural point of view amongst all models presented in this section.

An example of the application the ontology annotation model is shown in Figure 11. In this example, possibly different users annotated resources that represent the concepts of a `Person`, `Country`, and `Political Unit` as well as the entities `Barack Obama` and `USA`. For the annotation, the annotator(s) used relations `is-instance-of`, `is-president-of`, and `is-a` defined in some ontology specification. Other examples of the application of this model can be found in [9].

Pros Ontology-based annotations or "semantic annotations" describe a resource with respect to a formal conceptual model, allowing meaning-bearing links between structured and unstructured data (such as an ontology and a text). This empowers a whole new range of retrieval techniques, which can be based on the knowledge schema expressed in the ontology, benefit from reasoning, co-occurrence of annotation or entities in the same resource or context, as well as combine this with unstructured data specific types of retrieval, such as full text search (FTS) in information retrieval (IR). The actual metadata is encoded in the annotation and usually expresses metadata automatically or manually generated about the resource. The ontology and the corresponding instance base capture background knowledge about a domain. The combination of the evidence based information about the resource and the background knowledge, allows indexing techniques, which are based on resource URIs as modeled in the ontology, ensuring retrieval and navigation through each of its characteristics (for example lexical representations such as NYC and New York City will be indexed as a single resource, despite their superficial differences, and this will lead to results containing the string "New York City", even though the user provides a query such as "NYC").

Cons Each of the annotation models described in order of increasing complexity, presents new challenges to human annotators, although disclosing richer potential for automatic processing. Semantic annotations, being the most sophisticated of this row, are no exception. The main challenges semantic annotation presents are in two major lines, namely (i) usability, and (ii) maintenance of the conceptual models.

The usability aspect is fundamental to human involvement in the generation of semantic metadata and is also going to be the main hurdle that needs to be crossed to weave the approach in all forms of user interaction with software and data. Proposing a large number of entities and concepts coming from an ontology to the annotator is indeed an issue. There should thus be efficient search and recommendation services to help the annotator in providing the right entity or relation for the semantic annotation.

This raises another issue in the use of complex ontological structures. In fact, one can experience the challenge of presenting multiple subsumption structures over the same model, as well as multiple description facets of each resource. Empowering users to find their way to the right concept, entity or relationship that they want to cite is a serious challenge to usability experts and visual interface designers. In fact, as it is shown in [1], "*Most people find it difficult to understand the logical meaning and potential inferences statements in description logics, including OWL-DL*". In addition, in real world applications (in the Linked Open Data (LOD) cloud [12] for example) multiple instance bases and fact bases can be involved and thus, their corresponding ontologies can be inter-aligned [13], resulting in millions or even billions of individual entity descriptions.

While the issue of scalable user interfaces that can deal with large amounts of possible values for an annotation also arises with the previously described models (such as tags or attributes), the complexity of displaying the right entities and relations from multiple large aligned ontologies further complicates the usability problem and creates new challenges in terms of scalability. Indeed, while intuitive search and auto-suggest/complete methods can already help with simpler models, the complexity of the structure of the ontology raises new issues in displaying large amount of entities and their relations as well as in summarizing of entire knowledge bases to any useful level of granularity.

Another complicated task an ontology provider needs to face is the maintenance and update of the knowledge, often coming from external sources, its syntactic and semantic alignment, and often, its challenging scale (e.g. bio-medical knowledge bases with billions of individual facts). In [14], the authors study different automatic approaches to construct or extend ontologies, they conclude that most of the state of the art approaches require some level of expert involvement to curate the knowledge. This is also pointed out in [15].

Applications OntoWiki [16] is a free, open-source semantic wiki application, meant to serve as an ontology editor and a knowledge acquisition system (see Figure 12). It allows its users to annotate Web resources representing them as concepts and instances of concepts. Different attributes can be assigned to a resource and interlinked with other resources, therefore, describing its characteristics. OntoWiki retrieval functionality allows to search and to generate different views and aggregations of resources based on concepts, attributes and relations.

Semantic MediaWiki [17] is an extension of a wiki software[21] that allows its users to annotate wiki articles defining an article as a class like "person", "document" or an instance of previously declared class. The users can also interlink articles by annotating them with

[21] the same one as Wikipedia uses.

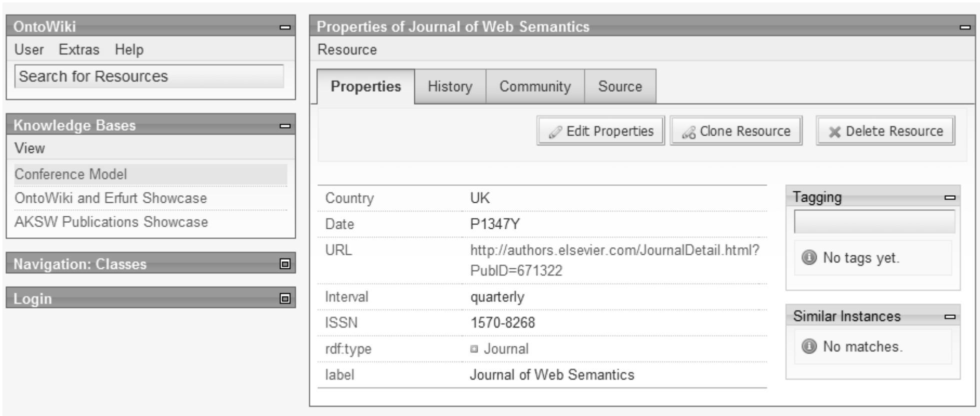

Fig. 12. Resource Annotation Using the Ontology Annotation Model in OntoWiki

typed links like “author” or “was born in”. This brings semantic enhancement to the MediaWiki that allows browsing and searching in a more specific ways, such as “Give me a table of all movies from the 1960s with Italian directors”. OntoBlog [18] takes a similar approach as Semantic MediaWiki, but is based on a blog platform and allows the authors of the blog posts to define attributes and relations to an ontology’s entities or other blog posts.

The authors of [19] describe a similar system that allow the annotation of wikis with semantic content; however, unlike the general entity/instance approach used by the systems described earlier, the annotations represent rules and knowledge that can be used for problem solving in expert systems.

While the previously discussed systems allow the content creators to dynamically create new class and instances in a centralized website, the Annotea [20] annotation scheme focuses on allowing content consumers to annotate web pages distributed over the web with a basic set of classes[22]. For instance, users visiting a website can leave comments, questions or explanations to the content provided by the content creator.

KIM is a semantic annotation, indexing and retrieval platform developed by Ontotext [21]. It can mainly be used to facilitate automatic semantic annotation on top of different content types, with built-in extended capabilities for semi- or non-structured text processing based on the GATE framework [22]. It is deployed on top of a native semantic database engine, currently OWLIM and/or Sesame [23,24]. It allows both of the above described kinds of semantic annotations, in the same time providing support for the simpler annotation models explained earlier and in the next section. Despite the fact it has matured as a platform since the early 2000s and is in active use, it has never, so far, been focused on usability or visualization aspects of the manual annotation process.

PhotoStuff [25] and PicSter [26] are also annotation systems based on ontologies, allowing to describe instances and classes within images (as their name suggest). While KIM allows to interlink text to the ontologies to describe instances, PicSter allows for the creation of instances of entities depicted in figures (e.g. a type of flower in a photo). It also allows to annotate the pictures with metadata attributes relevant to the picture such as where it was taken.

Microformats[23] [27] allow for the embedding of instances description directly in HTML, thus annotating the content with properties and relations to other entities on the web. hRest [28], for example, can describe RESTful web services directly from their documentation pages. The Schema.org[24] initiative, started in 2011, is a demonstration of how such an approach of embedding semantic annotations within HTML is reaching a mature stage. In fact, as part of this initiative, a number of large search engine companies[25] have agreed to work together towards a standardization of semantic annotation within HTML, in collaboration with one of the main web standardization body, the W3C, and its “Web Schema” task force[26].

[22]note that in the Annotea scheme, new classes can also be described, but in a less open manner.

[23]http://microformats.org/

[24]http://www.schema.org

[25]including Google and Bing!

[26]http://www.w3.org/2001/sw/interest/webschema.html

While there are many existing vocabularies to annotate content on the web (such as the Dublin Core[27], Common Tag[28], SIOC[29], etc.) with a formalised ontological structure, and many ways of implementing them (microformats, RDF, RDFs, NTriple, etc.), FOAF [29] is a popular example of an annotation scheme to describe entities found on the web, in particular people and organisations, describe their relationship and their relation to web resources (`maker` for instance). Anyone can embed FOAF description on their website to provide contact metadata (`dateofbirth`, `name`, etc.) and relations to other entities around the web by using dereferenceable URIs to other FOAF descriptions or resources descriptions.

3. The Vocabulary Type

When annotation elements (e.g., tags, attribute names and values, relation names) are provided by the user in the form of a free-form natural language text, these annotations unavoidably become subjects to the semantic heterogeneity problem because of the ambiguous nature of natural language [30]. Particularly, we identify the following four main issues:

Base form variation. This problem arises when the same word is entered (possibly by different users) using its different forms (e.g., plurals vs. singular forms, conjugations) or erroneously (e.g., misspellings) during the annotation or search [30]. This is usually dealt with a lemmatization procedure that converts the annotation to its base form.

Polysemy. Annotation elements may have ambiguous interpretation. For instance, the tag "Java" may be used to describe a resource about the *Java island* or a resource about the *Java programming language*; thus, users looking for resources related to the programming language may also get some irrelevant resources related to the Island (therefore, reducing the precision);

Synonymy. Syntactically different annotation elements may have the same meaning. For example, the attribute names "is-image-of" and "is-picture-of" may be used interchangeably by users but will be treated by the system as two different attribute names because of their different spelling; thus, retrieving resources using only one of the attribute names may yield incomplete results as the computer is not aware of the synonymy link;

Specificity gap. This problem comes from a difference in the specificity of terms used in annotation and searching. For example, the user searching with the tag "cheese" will not find resources tagged with "cheddar[30]" if no link connecting these two terms exists in the system.

The use of free-form natural language requires the minimal involvement of the user at annotation time but leads to a higher involvement of the user at the search and navigation time. The four problems described above produce a higher level of noise in search results and the annotation consumers need to filter out manually the results they are interested in. At navigation time, the lack of an explicit structure that defines the relationships between terms used for the annotation largely reduces the navigation capability and only simple clustering can be performed to help the user navigate through the annotations (such as the one illustrated in the Delicious tag cloud in Figure 4).

The problems described above can be addressed to a certain extent by using a Knowledge Organisation System (KOS) that can take the form of an authority file, a glossary, a classification scheme, a taxonomy, a thesaurus, a lexical database (such as WordNet [31]), or an ontology, among others (see [32] for an in-depth discussion on the different types of KOSs). The key idea is that terms in user annotations and queries can be explicitly linked to the elements of the underlying KOS and, therefore, their meaning can be disambiguated (see Figure 13). Depending on the type and structure of the KOS, the above mentioned problems can be addressed in a different extent, as discussed in the rest of this section.

The controlled vocabulary is a noteworthy example of a KOS and can be defined as "*a closed list of named subjects, which can be used for classification*" [33]. Controlled vocabularies are normally built and controlled a top-down fashion by a (small) group of experts in a particular domain. When using such a vocabulary, the terms used by the annotation creators can be linked to elements of the vocabulary (e.g., word, concepts) and thus be disambiguated (recall Figure 13). The same approach can also be applied to disambiguate search terms used by the annotation con-

[27] `http://dublincore.org/`
[28] `http://commontag.org`
[29] `http://rdfs.org/sioc/spec`

[30] which is a kind of cheese

sumers, thus solving some of the issues described earlier.

Noteworthy, the term *controlled vocabulary* comes from the Library and Information Science community and in the Semantic Web community the term *ontology* is often used in order to describe similar kinds of knowledge organisation systems.

Ontologies are often classified based on the level of their expressivity (or formality) that conditions the extent to which a certain form of ontology can be used in automated reasoning. The authors of [34] proposed such a classification shown in Figure 14 where they note that there is a point on the scale (marked as a black bar) where automated reasoning becomes useful: this is where the ontology can be reasoned about using the subclass relations [2].

Note that the way how ontologies can be used to *annotate* resources (as described in Section 2) is different from the way how ontologies can be used to *support* the annotation process (as described in this section). In the former case, users build ontologies by providing pieces of the knowledge it contains as annotation elements. For example, by linking a page about Barack Obama to a page about people with the ontological relation `is-instance-of` (recall the example depicted in Figure 11), the user annotates the page about Barack Obama with an ontology annotation element. Such elements (possibly provided by different users) are then assembled into a bigger ontology which can be seen as a complex ontological annotation structure used to describe the annotated resources. In the case when an ontology is used as a KOS to support the annotation process, the users provide (simpler form of) annotation elements and (semi-automatically) map them to the background ontology (see Figure 13). For instance, the annotator might tag a web page with the term "dog" and link it to the concept DOG in the background ontology. It is worth mentioning that both approaches can potentially enable automated reasoning.

In the rest of this section we discuss three types of annotation models which differ in whether they are based on a KOS and, if they are, in the kind of the KOS. In the selection of the models we followed the principle of staying within the spectrum of so-called *lightweight ontologies* [35]. The first model is not based on a KOS whereas the second and the third ones are. These last two models fall on the left and on the right side from the vertical bar shown in Figure 14 respectively. We show how the four issues described at the beginning of this section can be addressed in the last two approaches. Table 1 summarizes the semantic heterogeneity related issues and to which KOS types they apply.

Table 1
Semantic heterogeneity issues and their relation to types of KOSs

ISSUE	TYPE OF KOS		
	no KOS	**Authority File**	**Thesaurus**
Base form variation	√		
Polysemy	√		
Synonymy	√		
Specificity gap	√	√	

3.1. No KOS

In this model, annotation elements are not linked to KOS elements and, therefore, are subjects to at least the four problems described in the introduction to this section. In other words, free text is used for the annotation and search. Note that the search in this model is implemented as matching of strings representing query terms with those used in annotation elements.

PROS The advantage of this model is that the annotator and the annotation consumer does not need to know about the existence of a KOS and does not need to be involved to help resolve ambiguities in linking annotation elements to the elements in the KOS. Thus, the human involvement at annotation and search time is minimal: the user enters a list of free-text words in a text box.

CONS The main disadvantages of this model are the four problems described earlier: base form variation, polysemy, synonymy, and the specificity gap. Moreover, the minimal involvement at annotation time translates into a higher involvement required at the search and navigation time, as described in the introduction to this section. Folksonomies and other collaborative tagging systems based on free text annotations suffer from the lack of formal semantics that leads to the semantic heterogeneity problem, which makes indexing, search and navigation more difficult [36].

APPLICATIONS Delicious and Flickr are both examples of systems that use free text for the annotation and search. The user can enter any free-text tag they want to annotate their bookmarks and photos (respectively). Connotea [6] and Bibsonomy[31] (amongst

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

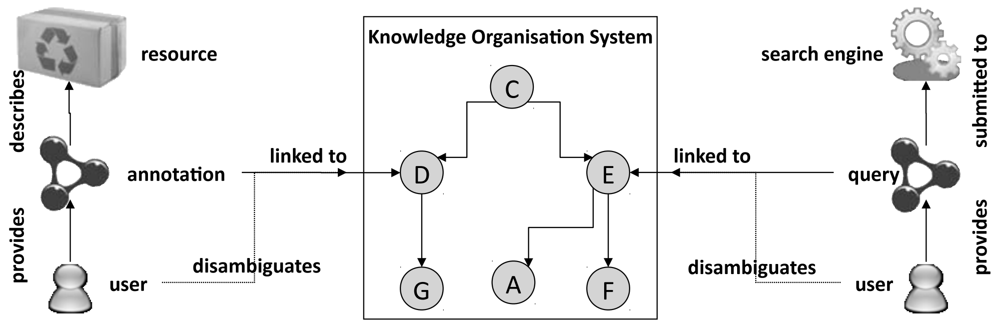

Fig. 13. KOS-based Annotation and Search

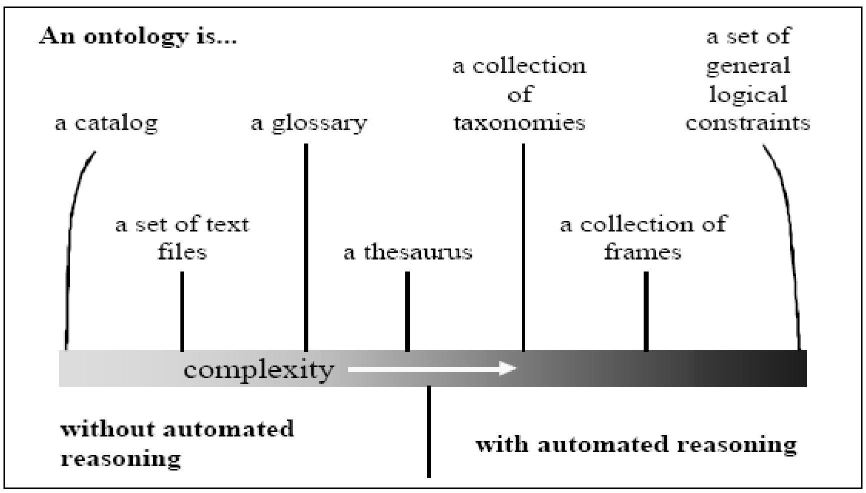

Fig. 14. A Classification of Various Forms of Ontologies According to their Level of Expressivity (Formality) (From [34])

other) allow for a similar type of free-text tagging but for scientific references.

3.2. Authority file

This model is based on a simple form of a controlled vocabulary called the *authority file*. In this vocabulary, synonymous terms are grouped to represent a concept and one of the terms is selected as the concept name and used for visualisation and navigation purposes (this term is called the *preferred term*) [33]. Each concept may have one or more associated terms and each term may belong to one or more concepts. In this model, the annotation elements are mapped to the controlled vocabulary terms and, consequently, to concepts which uniquely codify the meaning of the annotation elements. For example the tag “automobile” can be mapped to a concept with the preferred term “car” in the controlled vocabulary. Note that when a term belongs to more than one concept, the user may need to be involved to disambiguate the concept selection at annotation or query time.

Pros The model allows us to resolve the polysemy and synonymy problems by linking annotation elements and user queries to a concept in the vocabulary. For example, if the user searches for the term “java” that is mapped to a concept with the meaning of a programming language in the controlled vocabulary, the search will not return resources annotated with the term “java” that refers to the island in Indonesia. It is worth mentioning that the user can partially resolve the polysemy problem by adding more terms to the query. For example, querying with “java programming language” will unlikely return results related to the island. However, it does not address the problem of synonymy (thus affecting recall) and would require the user (as consumer of the annotations) to provide queries with more terms whereas user queries are typically very short[32].

Because synonymous terms are linked to a single concept in the authority file, the user (as annotation consumer) will be able to discover and navigate resources annotated with a given concept without having to know the particular terms the annotator and/or other annotators used to denote this concept while annotating these resources.

CONS At both annotation and search time, the user may need to get involved more. At annotation time, if there is an ambiguity in the mapping from the term to possible multiple concepts in the vocabulary, the annotator may need to choose which concept the term refers to. Similarly, at search time, if there is an ambiguity in the terms the users use for searching, they may need to specify explicitly which concept they are referring to. Alternatively, the user may decide to leave the concept disambiguation task to the system which may apply a disambiguation algorithm; however, this may lead to a large amount of erroneously selected concepts as the disambiguation task is known to be hard in the general case [37].

The second issue is the time and human resources needed to create and maintain the data in the authority file. Particularly, one of the most challenging maintenance problems is to keep the contents of the authority file aligned with the possibly emerging vocabulary of the users to cover their annotation and search needs.

APPLICATIONS Faviki[33] is a social bookmarking system similar to Delicious (see Figure 15). However, when users enter a tag, they are asked to disambiguate its meaning to a known concept. These concepts are automatically extracted from the contents of DBPedia [38]. This disambiguation allows a better convergence between the terms used for the annotation by different users as well as it provides better results at search time. In [39], the authors provide a review of such semantic bookmarking systems.

Delicious introduced a few features to its website to help the users put more control on their vocabulary. By adding the recommendation and auto-completion features to the tagging interface, they provide a dynamically extensible vocabulary where users can add new free-text tags or reuse the ones already existing in the community vocabulary. Apart from this, Delicious added the feature of creating tag "bundles" where users can group together their tags in any way they want, thus helping them disambiguate and control their vocabulary more strictly. Note that although authority files are created by domain experts whereas Delicious bundles are created by ordinary users, the two data structures are similar and, therefore, can be subjects to the same advantages and disadvantages as discussed in this section. We extend this discussion in Section 4, where we provide details on the different ways to create and maintain the contents of the underlying KOS (e.g., by users or by experts).

3.3. Thesaurus

A thesaurus is "*a vocabulary of a controlled indexing language, formally organized so that the a priori relationships between concepts (for example as "broader" and "narrower") are made explicit.*" [40]. Figure 16 provides an example of a thesaurus. In this thesaurus, the concepts "Cat" and "Dog" are linked to their parent concept "Mammals" with the genus-species relation which is then linked to a concept "Vertebrate" with the same relation.

Note that a thesaurus can be constructed by extending the authority file with the parent-child relationships between the concepts in the authority file. This thesaurus structure is the basis for the KOS of the model discussed in this section.

PROS This model inherits the pros of the model based on the authority file (see Section 3.2). Apart from that, this model allows us to solve the specificity gap problem because query terms can be mapped to the annotation elements through the parent-child relations of the thesaurus and, therefore, resources which are more specific in meaning than the user query can also be retrieved. For example, the user can annotate two different resources with "cat" and "dog" and then retrieve both resources by searching for "mammal".

CONS This model inherits the cons of the model based on the authority file (see Section 3.2). Another disadvantage of thesauri is the problem of the vocabulary granularity. Namely, if someone (e.g., a domain expert) produces a thesaurus, there is no guarantee that this one will be detailed enough for the user for the annotation and search. For instance, in the thesaurus example presented earlier (see Figure 16), the expert might not have introduced the "vertebrate" vs. "invertebrate" distinction. In this case, if the user wants to

[32] An analysis of the top 1000 queries submitted to the Yahoo search engine showed that the average length of a query is 1.8 terms (see `http://webscope.sandbox.yahoo.com/`).

[33] `http://faviki.com`

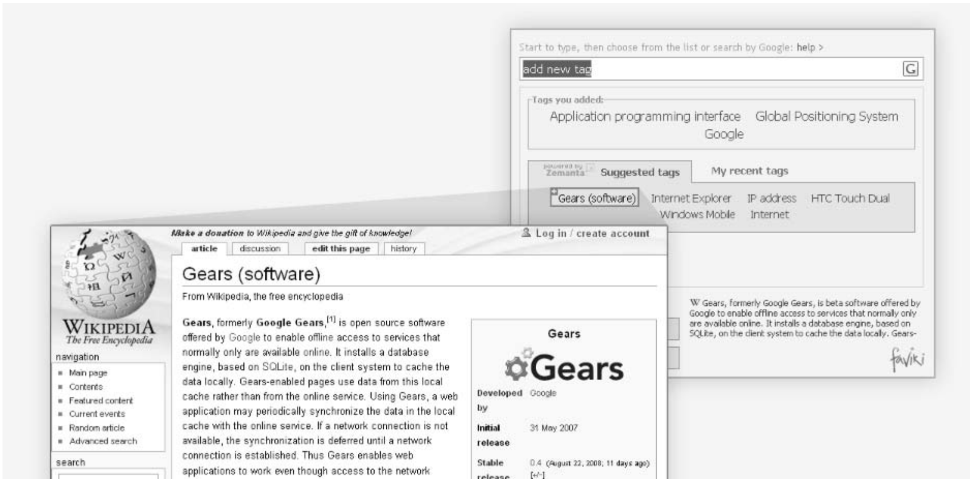

Fig. 15. Faviki: Tagging with Concepts from Wikipedia (screenshot provided by faviki.com)

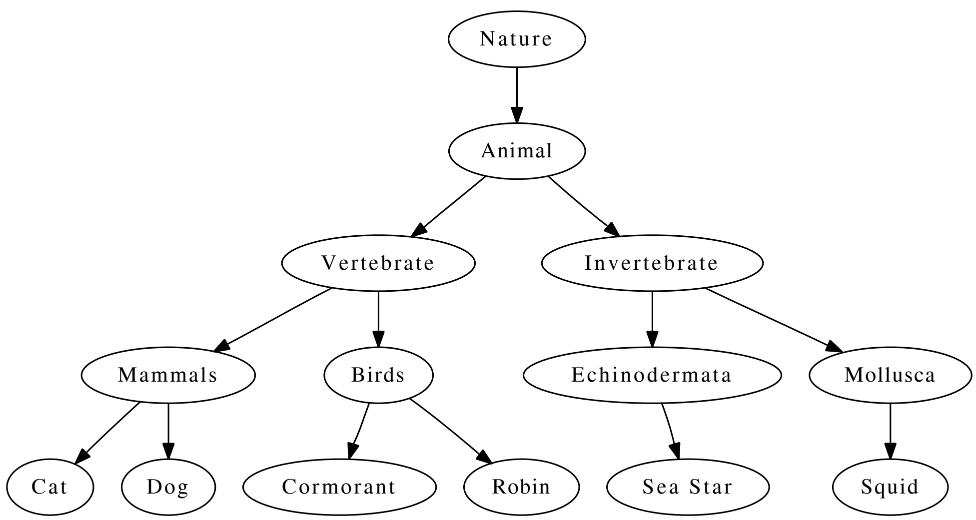

Fig. 16. Example of a Thesaurus

perform a generic search for all "vertebrate", only a search for the higher term "animal" will be possible. In contrast, if the thesaurus provided by the domain expert is too detailed, the disambiguation task at annotation and search time may become more complex for the user who might not know the differences between terms which are very close in meaning.

APPLICATIONS Shoebox[34] is a photo management software that allows the users to annotate photos with annotations which are nodes in a thesaurus. The software comes with predefined categories (such as countries/regions/cities, or animal kingdom) but the users can also freely extend the thesaurus by creating their own categories and sub-categories. The user can then search for all photos tagged with an annotation corresponding to a category or one of its sub-categories as illustrated in Figure 17.

The approach presented in [5] uses the SKOS vocabulary schema [41] to represent the thesauri used for annotating cultural heritage resources such as scanned book-printing in the Bibliopolis[35] collection.

[34] http://www.kavasoft.com/Shoebox

[35] http://www.bibliopolis.nl

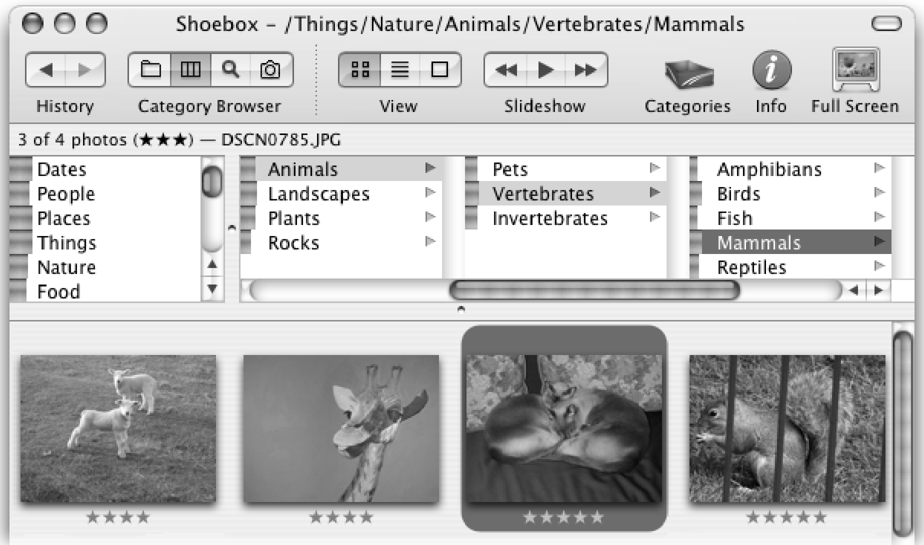

Fig. 17. Searching of Photos by Categories in Shoebox: Here Things > Nature > Animal > Vertebrate > Mammals

4. User Collaboration

In this section we will describe a dimension related to how users contribute to the creation of different types of annotations (see Section 2) and the vocabulary, or KOS, used in the process (see Section 3). We can distinguish between two approaches that can be applied for the creation of the annotation or for the creation of the KOS:

1. the single-user model, where a single user performs the task of either annotating resources or creating the vocabulary (or both),
2. the community model, where a set of users collaborate in the task of either annotating resources or creating the vocabulary (or both).

In this section we will focus on the interactions between users, and how these interactions affect, positively or negatively, the other two elements of our classification of annotations; these elements are the structural complexity of the annotation, i.e., how users interact with each other using some degree of structured annotation (see Section 2) to annotate resources, and the type of vocabulary used to annotate the resources (see Section 3). Note that a system can have combinations of the single-user and community approach with the annotation and the vocabulary or KOS building. For example, a system can use a community build vocabulary or KOS, and have only single user annotations.

Table 2

Types of Collaboration for Annotating Resources and Building Vocabularies

Feature	Type of annotation		
	Single User		**Collaborative**
	Private use	**Public use**	
Interaction between Users when annotating	none	none	encouraged
Type of vocabulary	personal, private	personal, shared	shared

Considering the annotation perspective, in the single user model we can identify two sub-dimensions by considering who consumes the annotations (see Table 2). A single user can annotate a resource for personal use only, or can annotate a resource considering that these annotations will be later consumed by many other users. As can be seen Table 2, in the personal use of annotations the vocabulary used for the annotations can be local, and the user is free to use any form of vocabulary as long as s/he understands the meaning of the annotation (e.g., the user can add an annotation "Fufy" and s/he might understand that this is the name of his/her pet). In contrast, in the public use of the annotations, the annotator should use some form of shared vocabulary, as the intention of the annotation is for others to consume, this holds also for the collaborative annotations. In the remainder of this section we will explain how this consideration affects the annotation process.

Considering the vocabulary perspective, i.e., how the vocabulary is built to be later used in the annotation process, we will focus on KOS type vocabularies (see Section 3.2 and Section 3.3), since in the free-form annotation model (No KOS, see Section 3.1), anyone is free to use any term (even terms not corresponding to a vocabulary).

4.1. Single-user (Private use)

In this model, a user annotates resources for personal purposes, usually to organize these resources for future search or navigation, i.e., the same user is the annotator and consumer. Single users could also build their personal KOS (see Section 3), but as we will see in the CONS part, the necessary work does typically not justify this task.

PROS The advantage of the *single-user annotation model for private use* is that each user is in full control of the annotations and since there is no sharing, the issues related to the semantic heterogeneity problem (see Section 3) is reduced to the scope of the single user. One can also argue that since the annotator and the consumer of the annotation are the same person, the annotations will reflect the personal taste/opinion and knowledge of the user, which in turn can be translated into more accurate results at search time.

CONS In this model, each single user has to annotate all the resources that s/he wants to have annotated for future use. These resources can be private resources (e.g., local files), resources shared by other users (e.g., shared photos), or publicly available resources such as Web pages. This annotation process has the following disadvantages:

- annotating all of the resources takes time and requires motivation (building personal KOS to annotate the resources requires even more time and motivation). If the user has no strong incentives, the quality and coverage of the annotations and the KOS cannot be guaranteed; Indeed [42] found that users rarely annotated their resources if they were for a private use but that if they annotated photos – on Flickr in this study –, it was to provide better search for other people.
- users have to remember which terms they normally use to denote a particular concept when annotating a resource and use it consistently across the whole set of resources. If this is not done properly the user will not be able to find the desired resources at search time (see Section 3). While this is also true for the collaborative model (see Section 4.3), given the higher number of users annotating the same resource we hypothesize that there is more diversity in the use of terms to denote the same concept, and therefore in the collaborative model, even if the user searches a resource with a different term than the one s/he applied in the annotation, this different term might be present too.

APPLICATIONS Bookmarking is a well known example of annotation of Web resources for personal use. Currently, most Web browsers include this feature by default.

Other examples of personal annotation systems are Picasa desktop[36] and Shoebox, where users tag their photos with free-form keywords. Picasa also adds the possibility of geo-tagging photos while Shoebox provides a KOS in the form of thesaurus for the annotation.

Another system for the annotation of Web resources is Zotero[37], where users have the possibility to manage Web resources related to scientific research by classifying them and/or by adding free-form annotations.

4.2. Single-user (Public use)

In this model, a single user, (normally a *classification* expert, or a small set of experts representing an institution) annotates resources with the goal of organizing the knowledge for a broader set of users that will consume these annotations. Library catalogs are perhaps the most well known example of this kind of model.

Considering the construction of the KOS, in this model also a single user (also normally a *domain* expert or a small set of experts representing an institution) builds the KOS to be used by a broader set of users in the annotation task (normally, but not only, the *classification* experts mentioned above) and later for search (normally by the public in general). In this

[36] http://picasa.google.com/
[37] http://www.zotero.org/

case, library classification schemes or thesauri are well known examples.

Note that the expert, *classification* of *domain*, have different tasks and therefore can be performed by different people. Classification experts are experts in the annotation task, they classify by annotating resources with the predefined classes in a predefined vocabulary, and domain experts defined the classes and the vocabulary to be used in the classification and annotation task. There are cases, however, when the *classification* and *domain* expert role is performed by the same person. Whenever we mean to refer to only one type of expert we will make the clear distinction.

PROS Many people benefit from the work performed by a few experts in the field (both classification and domain); the work (either the annotation or the resulting KOS) is considered to be of good quality resulting in a good organization of the resources. In addition, experts in the annotation and classification task are also able to use more complex and well structured vocabularies (see Section 3) as they are better trained in this task. Well studied and consistent KOS in the form of controlled vocabularies developed by domain experts are very useful for solving the issues related to the semantic heterogeneity problem (see Section 3).

CONS It is normally costly to have dedicated experts to annotate and classify resources or to build KOS in the form of controlled vocabularies. Moreover, in highly dynamic domains, the time lag between the publication of the resource and its annotation can be considerable [43]. This delay can be due to, either the scalability of the approach (work overload of the classification experts) making the classification of vast amounts of resources in short times impracticable, or to the fact that the field of knowledge is new or very dynamic and thorough studies have to be performed in order to reach consensus on the right vocabulary to be used for the annotations.

Another issue with this approach is that the classification experts who perform the annotation are neither the authors of the resource nor the consumers of the annotations, therefore the annotations might not always reflect the consumers' perspective or the authors' intention [44]. Some possible solutions to these issues, still at research stage, are being proposed; for example, to let casual annotators define their classes and keywords that will later be approved by domain and/or classification experts [45].

If the single annotator is not an expert in the annotation and classification process (or vocabulary building) it could miss relevant annotations (or terms) for a resource. This is also true in the collective annotation model (see Section 4.3) but given the higher number of users annotating the same resource (or building the vocabulary) we hypothesize that there is more probability that more meaningful annotations will be used given that the resource is analysed from possible different perspectives.

APPLICATIONS Library catalogs, such as the Library of Congress[38] and many other libraries fall into this model of annotation. Library classifications schemes used in library catalogs are examples of expert designed KOS in the form of controlled vocabularies with the purpose of classification of books.

4.3. Collective annotation

In the collective annotation model, a set of users share their annotations about publicly available resources. These annotations are later also used by a set of users (possibly larger than the set of annotators) for navigation and search. In this model, the workload of annotating resources is distributed (in contrast to the previous 2 models). The users do not necessarily work together in the process of annotation, neither do they have to reach an agreement on the resource annotations (but obviously this could be the case).

In this model, we could devise two subcategories, one where users collaboratively modify the same annotation (wiki style) and another where users can only modify their own annotations (folksonomy style). While in both subcategories users (annotators) might have different motivations to create or modify the annotations, the *pros* and *cons* to be presented here generically apply to both subcategories, unless explicitly stated otherwise.

Collaborative Tagging (or social tagging) is a well known model to annotate resources[39] with free-text tags [30]. When a critical mass of user-generated tags is available, these annotations can be used to generate folksonomies [46]. The main characteristic of folksonomies (folk + taxonomies) is the bottom-up approach (i.e., from individuals to groups) for reaching consensus on emerging concepts, without any a priory agreement or KOS.

Considering the construction of KOS, the basic idea is to let users collaboratively build a the KOS us-

[38] `http://www.loc.gov/index.html`
[39] mainly on the web.

ing a bottom-up approach. This approach is reflected in several proposals in the literature. These proposals differ in whether there should be a pre-defined KOS which will be collaboratively extended [47], or the KOS should emerge bottom-up from scratch [2,43]. Depending on the model, users could work together in order to reach agreements in the construction process [43], or they could work independently and an automatic process could address the evolution of knowledge [2]. [48] proposed several generic characteristics and issues that any system for *emergent semantics* should consider and address. The notions presented in these proposals allows us to address disadvantages present in centrally built KOS.

PROS The main advantage of the collaborative annotation model and construction of the KOS is the distribution of the workload among all annotators of the system. This, in turn, lowers the barrier of user participation in the annotation process since one user can adopt the annotations that were assigned to a resource by others — thus simplifying the annotation task —, or even, more importantly, can search and discover new resources that were annotated with the same tags as the user's tags [49]. Another advantage (in the folksonomy style) is that annotators can express their own views when annotating resources, which in contrast with annotations made by experts, could simplify search and discovery of resources for other users (consumers) with the same interests, considering that these resources were annotated with many potentially different points of views and interests.

In addition, the mass of data provided through the annotations in a collaborative model as well as the users' relations with these annotations can be used for automatic extraction of new knowledge (new terms and relations in the KOS); for example, a recommendation system can be built to propose relevant annotations for a new resource, based on existing annotations of similar resources [50] or to extract co-relations from unstructured annotation schemes [51,52,2,14]. While such techniques can be applied with any type of collaboration model, the state of the art in recommender systems use knowledge from multiple users as it is difficult to provide rich recommendations when only a single user is involved. In addition, these statistical techniques require a large amount of data to be accurate and while, theoretically it is possible to apply them on single-user annotation models, the critical mass of annotation is often practically only reached in multi-user systems.

This model also provides the system with behavioral information about the users' interests through their interaction with other users' annotations (consuming them) and their own annotations (producing them). By using this information the system could infer a user model containing, for example, their education, background, interests or culture. Based on this user model the system can suggest and discover related resources or can discover hidden communities and sub-communities of users based not only on social relations (such as friends or coworkers) but also on interests and shared knowledge or background [2]; Ebay and Amazon[40] are systems that use these techniques to suggest their consumer related products.

[37] illustrates how a static vocabularies or KOS might not be well suited for a dynamic online use as new terms arise quickly. Building the KOS in a bottom-up fashion can address the time lag problem for the inclusion of new concepts in the KOS. Theoretically, having more semantics in the annotations will enable better services and this also goes with the increase of the underlying vocabulary to match the whole vocabulary used by the user [48].

CONS Social annotation approaches could suffer from subjective annotation that express personal opinions, likes or dislikes of the annotators, and not the actual intention of the resource author. Examples of these subjective annotations are "best paper", "funny", "boring", "to read" [30]. This type of annotations cannot be avoided since the primary goal of any annotation model is to allow users to find the resources they are interested in. These subjective annotations could be also important for ranking purposes of the resources and the annotating users. The system could try to limit such subjective annotations by automatically separating the personal annotation from the general annotations or by avoiding the mix between annotations that express opinions and the ones that are not associated to the annotator's relative taste. In the folksonomy model each user will express their own preferences as stated before, but in the wiki model where users have to override other user's annotations, this could even become an editing war were each user (or groups of users) tries to imposed their own opinions[41] [53].

An important issue which remains open and needs to be treated in the community-built KOS is the model

[40] http://www.amazon.com/

[41] http://en.wikipedia.org/wiki/Wikipedia:Edit_warring

to be used to allow knowledge to emerge from the contribution of each single user. The process of agreement (either automatic or user driven) in knowledge evolution is still an interesting research issue to be addressed [48,43].

Applications Diigo[42] and Delicious are examples of applications implementing this model of annotation to build a shared collection of bookmarks. While they both allow for private bookmarks[43], their main features are built around the sharing of annotations with the site community to construct a public folksonomy.

Flickr also allows users to comment resources, tag or add them to their own personal favorite list. These tags are later used for search and navigation. [36] proposed and extension for searching Flickr content where WordNet [31] is used to expand queries, using this approach users are able, for example, to find results about "automobiles" when searching for "cars".

The work presented by [54] incorporates the ideas of collaboration to library catalogs, where the users can become more than simple annotation consumers by annotating resources in digital libraries and become annotation providers. Facetag [55] is another initiative following this direction. The rationale in this system is to try to integrate top-down (i.e., from experts to individuals) and bottom-up classification in a semantic collaborative tagging system, incorporating the ideas of faceted classification, which helps users organize their resources for later search, navigation and discovery. Facetag incorporates the idea of collaborative annotation and collaborative KOS in a single system.

Considering community-built KOS, [43] proposed an ontology maturing process by which new knowledge is formalized in a bottom-up fashion going through several maturity stages: from informal emergent ideas using input from social annotation systems such as folksonomies, to formal lightweight ontologies via a learning process involving users. They also introduced Soboleo[44], a system for collaborative ontology engineering and annotation.

Another related work to community-built KOS was presented by [2] where a model for formalizing the elements in an ontology evolution scenario was proposed. The model consists of three elements, Actor-Concept-Instance. This model has a straightforward parallel to the model adopted in section 1 to explain the annotation process as seen in Figure 1; where Actor is equivalent to *users*, Concepts to *annotations* and Instances to *resources*.

Tagpedia [47] tries to build a general purpose KOS to overcome the problems of Wordnet and Wikipedia. The main issues with WordNet are the lack of knowledge about entities (specially people) and the lack of support for incorporating new knowledge. Wikipedia contains a lot of information about entities but suffers from the lack of a more formal/ontological structure [47]. The idea of Tagpedia is to initially mine Wikipedia to construct an initial set of *syntags* (in contrast to synsets in WordNet that groups words) and to allow users to extend this initial set of *syntags* dynamically in a bottom-up manner.

In PicSter [26], image annotations are created locally by the users and thus there is no central ontology shared by all peers using the system. PicSter still provides the opportunity to share resources and their annotation by providing ontology alignment [56] when the users perform search queries.

Annotea [20] takes an interesting approach in collaborative annotations of resources in that it is decentralised. In particular, the resources (web pages) that are annotated are not always owned by the annotation creators and the annotations can be shared and distributed on different annotation servers. Annotation creators can leave comments, questions, etc. on websites that can be consulted by other users of the Annotea system and extended (with answers, explanations, etc.) by these separate users.

5. Discussion

As we have seen in the previous section, there are many approaches to annotation systems, from theoretic models to actual end-user implementations of such systems. While there have already been reviews of the state of the art in this field (e.g., see [57]) and a number of research workshops ([58,59] for instance), we are not aware of any approach that tried to characterize diverse semantic annotation approaches in a common framework. In the previous sections, we have thus tried to identify the main dimensions that would help us classify a large variety of systems together to show their common features. Table 3 shows a summary of some systems that we have discussed and how they fit in the different dimensions. In this section we discuss some properties and drawbacks of the

[42] `http://www.diigo.com/`
[43] thus falling in the *single-user (private user)* dimension.
[44] `http://www.soboleo.com/`

Table 3

Examples of Systems and their features according to our classification (* see comments in Section 5)

	Structural Complexity				Vocabulary Type			Collaboration Type		
	Tag	**Attribute**	**Relation***	**Ontology**	**No KOS**	**Authority file***	**Taxonomy**	**Single user, private use**	**Single user, public use**	**Collaborative**
Flickr	√				√				√	
Delicious	√				√			√		√
Faviki	√				√		√	√		√
Last.fm	√				√					√
Youtube	√				√				√	
Facebook		√	√		√	√			√	√
Ebay		√					√		√	
CiteULike	√	√			√					√
Bibsonomy	√	√			√					√
Zotero	√	√	√		√			√		
Mendeley	√	√	√		√			√	√	√
OntoWiki				√	√					√
Semantic Media Wiki				√	√					√
Annotea				√	√					√
Picasa Desktop	√	√	√		√			√		
Picasa Web	√	√	√		√				√	
Picster				√			√		√	

proposed classification as a whole (Section 5.1) and provide some recommendations for building semantic annotation systems based on these dimensions (Section 5.2).

5.1. Classification Properties and Drawbacks

Independance of the Dimensions In the previous sections, we have introduced three main dimensions to classify annotation systems:

1. Structural Complexity of Annotations,
2. Vocabulary Type,
3. User Collaboration.

While we have discussed these dimensions as being mainly independent – as, in our opinion, they are orthogonal –, we do not intend them to be considered separately as one cannot design an annotation system without making choices along each dimension. As we have discussed in the Vocabulary Type (Section 3) and in the Collaboration Type (Section 4), some aspects of these dimensions might be preferable in combination. For instance, when using large collaborative systems, it is often easier to use a simpler structural complexity, as Flickr or Delicious have done, as this simplifies the sharing of annotations and the user interfaces. However, as we have discussed, by using simpler structural complexity, one looses much of the semantics of the annotations and thus it might still be preferable to go for more complex structures and use more advanced annotation alignment algorithms as is illustrated by [26]. Another example of this preferential tie between dimensions is the one of the ontology structural complexity. When using such annotation system, the annotators will create instances and classes while annotating. It seems thus natural to allow the future annotators to reuse this created ontology as a vocabulary for their own annotations. While this is not a real dependence between the structural complexity and the vocabulary type – i.e. the vocabulary available could still be a No KOS –, it probably makes more sense to provide such features to the user. In these cases, the dimensions remain orthogonal and independent but display some pairing preferences, and this is why in the previous discussion some systems were found in many dimensions.

Structural Complexity In addition, there are some fuzzy separations between the possible sub-dimensions of each main dimension. For instance, Flickr and Delicious are both marked as having a *tag* structural complexity in Table 3. However, users have started using the tag system as a workaround to store more complex data by using machine tags; thus being able to store attributes and relations in the tags. We do not directly consider these systems as supporting attributes or relations as, at the implementation level, when dealing with machine tags stored as free-text tags, it is very hard to leverage the full semantic services provided by the attribute or relation dimensions of the structural complexity. Indeed, both Flickr and Delicious have had to provide specific support for machine tags, thus making them move away from purely free-text tag systems. For instance, Flickr eventually introduced a full-fledged geotagging features, separate from the tags annotations, after many users started using machine tags to store latitude and longitude of photos.

Another important point to consider is the difference between attributes and relations in the structural complexity. When a resource is represented as a URI in an annotation system, it is then easy to encode a relation as an attribute, which domain is a URI. Thus, at a data storage level, the attribute dimension and the relations dimension are not really different. However, knowing that a particular attribute represents a relation can enable more powerful features. For instance, if for a person, we have one attribute “picture” that points to the URI of the file and one attribute “father” that points to the URI of another person. If they are both only attributes, there is no difference for the computer as to what to do with both fields. However, if the “father” attribute has the semantics of a relation, then the computer might know that there is more to this URI than just a reference. For instance, mutual relations could be computed, possible transitivity could be computed, etc. Thus relations are still interesting to consider, and we have kept them as a standalone sub-dimension as it provides a larger feature-set.

Vocabulary Type The same issue arises when we consider the vocabulary type. While many systems use the No KOS approach – for instance Flickr free-text tagging –, some still provide an auto-completion or recommendation feature that helps users choose tags from a list of popular tags (for example). If we consider this, systems with such a feature – such as Delicious – are standing between the No KOS dimension and the use of a simple authority file made of the most popular tags. However, as for the structural complexity, while this approach provides support for the user when creating annotations, it might not provide all the features of an authority file; for instance there is no control over the base form variations of the tags.

Thus, while the top level of our classification is meant to represent orthogonal dimensions, the second level is meant to be taken as main divisions between feature sets, which can overlap at some time to provide hybrid systems as with the machine tags or the auto-completion of tags. However, as we pointed out, this overlap also has its drawbacks as it does not enable all the features of the more complex dimension. We have still pointed out along the article when these overlaps between approaches existed while trying to keep a logical split between the different level of each dimension.

5.2. *Recommendations for building semantic annotation systems*

In this section we provide our recommendations for the configuration of a balanced semantic annotation system that, on one hand, encapsulates enough semantics to allow the development of new semantics-based services for the end-user and, on the other hand, does not require an explicit expertise from the user to provide semantic annotations, which would facilitate a wider adoption of such systems. Our recommendations are based on the analysis and findings reported in this article as well as our experience with the INSEMTIVES project annotation systems (see Section 6) and are provided for each of the three dimensions, as reported below:

Structural Complexity of Annotations. While users should be able to provide annotations of the tag, attribute, and relation kinds (see Section 2), they should be driven towards providing more relations and attributes to enable more semantics-based services. In this case, the user will provide qualified resource descriptions and will interlink resources with qualified links. This results in a structure that can be naturally mapped into RDF and, therefore, can enable the adoption and reuse of services developed by the Semantic Web community so far. In addition, these lightweight but structure-rich annotations can form the basis for the automatic extraction of data sets to be uploaded to the LOD cloud. The use of the full fledged ontology for annotation is not recommended when the annotating user is not an expert in semantic web technologies due to high learning barriers [1].

Noteworthy, the tendency towards more structured annotations can be noticed in some annotation systems. For example, while Flickr and Delicious only started with free-text tags, they soon introduced more complex attribute annotations: both Flickr and Delicious provided support for attributes and relations based on machine tags; Flickr also introduced a dedicated geo-localisation interface to store the coordinate (attributes) where a photo was taken; Delicious introduced a way to send a resource to other users by linking to them with the `for:` tag. In the same way, Wikipedia (and MediaWiki) introduced info-boxes that allows to annotate wiki pages with attributes and this feature was then extended by projects like Semantic MediaWiki [17].

Vocabulary type. We recommend that, during the annotation process, the user is supported by a KOS of the thesaurus type. This will enable one of the mostly used reasoning tasks, the subsumption on the hierarchy of concepts (see Section 3.3) and will allow to address the problems described at the beginning of Section 3. While the annotating users should be driven towards providing annotations that are linked to KOS elements, they should not be restricted from providing free text annotations. This recommendation is to support the annotators who are well familiar with the existing Web 2.0 annotation systems and to overcome the potential problem of a partial coverage of the KOS, which can be addressed, at least in part, by adopting the recommendation for the user collaboration dimension.

In public annotation systems, we can see that this trend towards more structured vocabularies is growing, with the introduction of tag "bundles" in Delicious, with the use of categories in Wikipedia or with new social bookmarking sites based on KOS such as Faviki.

User Collaboration. During the annotation process, whenever possible, it is recommended that the workload of annotating resources is distributed among the users of the system. This is to leverage the "wisdom of crowd" effect [60] – the concept that has been successfully implemented in many Web 2.0 applications such as Delicious. Considering the KOS, it is recommended that the contents of the KOS are evolved dynamically by the users in order to address the problem of partial knowledge and time lag between the creation of concepts inside communities and their inclusion in the KOS by experts. From the user perspective, this means that they are more in control of the vocabulary, but from the system design perspective, this means that new methodologies for evolving knowledge are required. In order to avoid the "cold start" problem, it is recommended to provide an initial data in the KOS (possibly provided by experts). Because the users will have to deal with the semantics of annotations, proper user interface and interaction tools need to be developed that would expose to the user these semantics in a simple and natural way. For example, when the user enters "java" as a tag annotation, the system can show the summary of the recognised meaning of the tag as, for example, in "java (island)" that would help the user understand if the meaning is the intended one and correct it otherwise [61].

6. Using the Annotation Model Dimensions

The three classification dimensions that we discussed in the previous sections are not only useful to classify the existing annotation systems but also to specify requirements for developing new annotation systems. While different technologies can be used to implement a particular annotation model (e.g. RDF, microformats, relational databases), the system's high level features are not dependent on such technology and there is no need to provide such low level details to the users.

Noteworthy, the proposed classification was used within the INSEMTIVES project[45] in order to identify the requirements for annotation models in three different industrial use cases. In this section, we describe the methodology that was applied for this analysis (Section 6.1) and provide details of such an analysis for one use case partner who is a big player in the telecommunication sector (Section 6.2).

6.1. Applied Methodology

In order to identify the needs of the use case partners for the different characteristics of the annotation model, a semi-structured interview was performed with a representative of each use case partner. Before the interviews, each use case partner was given the report on the three dimensions of annotation systems (as presented in the Sections 2, 3 and 4). This allowed us to increase the awareness of the use case partners of the existing systems and their features and helped the partners to make more informed decisions about their needs. About forty questions were prepared for the guided interviews[46]. The questions were made as nontechnical as possible but with clear connections to the elements of the annotation model they referred to. Below we give some examples of the questions with an explanation of their relation to the classification dimensions:

Structural Complexity

> *Should the user be able to explicitly refer to properties of the resources the user query refers to?*

RATIONALE: the goal of this question is to clarify if the attribute structural complexity should be used.

> *When searching for a resource, should the user be able to find resources that have a certain relationship with another resource?*

RATIONALE: the goal of this question is to clarify if the relation structural complexity should be used.

Vocabulary Type

> *Should the user be able to navigate from a given resource to other semantically related resources?*

RATIONALE: the goal of this question is to clarify if the annotations should be assigned explicit semantics.

> *If they navigate, are they interested in seeing a detailed classification/thesaurus (such as by country, then by region, then by city, etc.) of the resources or a rough clustering is enough (such as the tag cloud)?*

RATIONALE: the goal of this question is to clarify if the thesaurus KOS should be used.

Collaboration Type

> *Can the set of available annotation terms be extended dynamically (or automatically), and by whom?*

RATIONALE: the goal of this question is to clarify if there is a need of updating the vocabulary, and whether collaboration is needed or allowed.

[45] FP7-231181, see `http://www.insemtives.eu`

[46] Interested readers are referred to [62] for the complete list of questions and for the transcripts of the interviews.

Will the same people that make the annotations, search based on the annotations? Are annotators and end user different?

RATIONALE: the goal of this question is to clarify the annotations need to be shared, or if the annotations should be private.

The answers to all the questions were then rigorously analysed with the goal of the identification of the requirements for the annotation model for each use case partner as reported in [62]. Once the requirements of each use case partner were mapped to the three classification dimensions, the implementation technology was decided, based on specific particularities of the current infrastructure of the use case partner (e.g,. use an RDF or a relational database).

6.2. Case Study – Telefónica Corporate Portal

Telefónica Investigación y Desarrollo[47] (Telefónica I+D for brevity) is the innovation company of the Telefónica Group[48]. Over the last few years and within the global market, Telefónica I+D has grown to become a network of centres of technological excellence that stretches far beyond the Spanish borders, extending its R+D activities to offices located in Spain (Barcelona, Granada, Huesca, Madrid and Valladolid), Brazil (São Paulo) and Mexico (Mexico D.F.). Currently, Telefónica I+D employs more than 1 200 employees shared out amongst its 7 centres.

The Telefónica I+D's intranet corporate portal is a live, 24x7, highly active Web portal used by every single employee at the company. Since its first days in 1998, the intranet corporate portal has become one of the main communication channel available at the company. The portal offers access to all the sources of information and services; including infrastructure management, business management, knowledge management and human resources management information, services and tools. However, as the amount of data has grown, it has become much more difficult to access the right asset in the precise moment when it is needed.

[47] `http://www.tid.es/en`

[48] The Telefónica Group is a Spanish multinational group of companies operating in the Information Technology and telecommunication domain. The Group stands in the 5th position in the telecommunication sector worldwide in terms of market capitalization, the 1st as an European integrated operator and the 3rd in the Eurostoxx 50 ranking, composed of the major companies in Europe (June 30th 2010).

In order to address the above mentioned problem, it was decided to enrich the corporate portal with automatic, semi-automatic and user-guided semantic annotation capabilities in order to enable enhanced features such as semantic search, faceted navigation and intelligent recommendations. In the following we describe the requirements analysis along the three classification dimensions that led to the development of a semantic annotation component for the Telefónica portal.

6.2.1. Structural Complexity

The use case requires a mixed complexity in the annotation model. Users are only interested in an *easy and quick* annotation procedure and the annotators are already used to a *simple tagging interfaces*.

However, in the use of the annotation for searching and navigating, the users should be able to *filter out* searches by using particular attributes (e.g., the author, a date range, etc.). There is also interest in having an annotation model that supports relational annotations to link resources together, but also parts of resources for navigating between them.

The annotation model should thus be able to support *tags* which are not associated to any particular property of the resource, but also *attributes* to describe specific metadata and *relations* between (and within) resources for navigation purposes.

6.2.2. Vocabulary Type

During the interview, users made clear that the annotation should not be limited to a specific vocabulary and that users will enter free-text annotations. However, after further discussion and by looking at the intended usage, it appears that the free-text annotation will be combined with a KOS of some sort, provided by domain experts. The KOS might take the form of a thesaurus to ease the navigation and search, and the users will be ready to align their free-text annotations to the concepts in the KOS.

A set of expert users, with a good knowledge of the domain as well as of the annotation system, will provide the initial contents for the KOS that can be used to bootstrap the recommendation system. This KOS will then be extended by the users when providing free-text annotations.

There is also a strong requirement that the issues of synonymy and polysemy should be resolved for improving the search accuracy. This will either require an *authority file* or *thesaurus* KOS. Hence, this use case will require a KOS where users will help disambiguating the annotation to map *free-text tags* to the corresponding concept in the KOS when performing the annotation task or searching.

6.2.3. Collaboration Type

The use case will have many shared resources that every user can access, annotate, and search for. With the exception of some of the resources where there is a need for access control, all the resources and annotations will be visible to everyone.

Moderators will be controlling the user's annotations to make sure they are correct, where traceability of the annotations is needed for auditing purposes. The collaborative annotation model thus should embed provenance information as well as versioning information.

In addition, as pointed out previously, the KOS will be built both in a *top-down* fashion, with a small set of domain experts creating the initial vocabulary but also in a *bottom-up* fashion, with annotators and moderators extending the existing KOS and annotating resources on their own.

7. Conclusions

This article investigates existing models for representing annotations, and analyses their different characteristics, forms, and functions. Based on this analyses, a classification for annotation models was developed that distinguishes three main dimensions:

- structural complexity of annotations,
- the type of the vocabularies used,
- the collaboration type supported.

Furthermore a comprehensive overview of existing annotation approaches, both in research and industry, is given to provide examples of annotation models and their different characteristics. Based on the analysis of the studied approaches and on the proposed classification dimensions, the article proposes a discussion on some inter- and intra-dimension characteristics of the proposed classification and on some recommendations of the authors for each dimension that can be useful to the designers of semantic annotation systems.

The article further reports on a methodology to use this annotation model classification scheme to elicit requirements for annotation models in end-user applications. This methodology is then illustrated with one concrete use case from which requirements for its annotation model have been extracted.

Acknowledgements

This work has been partly supported by the INSEMTIVES project (FP7-231181, see `http://www.insemtives.eu`).

The authors would like to thank Denys Babenko, Tobias Bürger, Borislav Popov, and Biswanath Dutta for their valuable contribution and feedback made at early stages of this work.

References

[1] Alan Rector, Nick Drummond, Matthew Horridge, Jeremy Rogers, Holger Knublauch, Robert Stevens, Hai Wang, and Chris Wroe. OWL pizzas: Practical experience of teaching OWL-DL: Common errors & common patterns. In Enrico Motta, Nigel Shadbolt, Arthur Stutt, and Nick Gibbins, editors, *Engineering Knowledge in the Age of the Semantic Web*, volume 3257 of *Lecture Notes in Computer Science*, chapter 5, pages 63–81. Springer, 2004. doi:10.1007/978-3-540-30202-5_5.

[2] Peter Mika. Ontologies are us: A unified model of social networks and semantics. *Web Semantic*, 5:5–15, March 2007. doi:10.1016/j.websem.2006.11.002.

[3] Flickr. Flickr blog. `http://blog.flickr.net/en/2010/09/19/5000000000/`, September 2010.

[4] David K. Gifford, Pierre Jouvelot, Mark A. Sheldon, and James W. O'Toole, Jr. Semantic file systems. In *Proceedings of the thirteenth ACM symposium on Operating systems principles*, SOSP '91, pages 16–25. ACM, 1991. doi:10.1145/121132.121138.

[5] Anna Tordai, Borys Omelayenko, and Guus Schreiber. Semantic excavation of the city of books. In Handschuh et al. [63].

[6] Ben Lund, Tony Hammond, Martin Flack, and Timo Hannay. Social Bookmarking Tools (II): A Case Study - Connotea. *D-Lib Magazine*, 11(4), April 2005.

[7] Glenn Rowe and Chris Reed. Argument diagramming: The araucaria project. In A. Okada, S.J. Buckingham Shum, and A. Sherborne, editors, *Knowledge Cartography*, pages 163–181. Springer, 2008.

[8] Bill Mann. An introduction to rhetorical structure theory (RST). `http://www.sil.org/`, 1999.

[9] Atanas Kiryakov, Borislav Popov, Ivan Terziev, Dimitar Manov, and Damyan Ognyanoff. Semantic annotation, indexing, and retrieval. *Elsevier's Journal of Web Semantics*, 2(1), 2005.

[10] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[11] Web Ontology Working Group. W3C recommendation for owl. `http://www.w3.org/TR/2004/REC-owl-features-20040210/`, February 2004.

[12] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data – the story so far. *Special Issue on Linked Data – International Journal on Semantic Web and Information System*, 5:1–22, 2009.

[13] Jérôme Euzenat, Adrian Mocan, and François Scharffe. Ontology alignments. In Martin Hepp, Pieter De Leenheer, Aldo De Moor, and York Sure, editors, *Ontology Management: Seman-*

tic Web, Semantic Web Services, and Business Applications, volume 6 of *Semantic Web and Beyond*. Springer, 2007.

[14] Andres Garcia-Silva, Oscar Corcho, Harith Alani, and Asuncion Gomez-Perez. Review of the state of the art: Discovering and associating semantics to tags in folksonomies. *Knowledge Engineering Review*, 26(4), December 2011.

[15] Katharina Siorpaes and Elena Simperl. Human intelligence in the process of semantic content creation. *World Wide Web (WWW) Journal*, 13, 2010.

[16] Sören Auer, Sebastian Dietzold, and Thomas Riechert. OntoWiki - A Tool for Social, Semantic Collaboration. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*, volume 4273 of *Lecture Notes in Computer Science*, pages 736–749. Springer, 2006. doi:10.1007/11926078_53.

[17] Markus Krötzsch, Denny Vrandečić, Max Völkel, Heiko Haller, and Rudi Studer. Semantic Wikipedia. *Journal of Web Semantics*, 5(4):251–261, 2007.

[18] Aman Shakya, Vvilas Wuwongse, and Hideaki Takeda. OntoBlog: Linking ontology and blogs. In Handschuh et al. [63], pages 47–54.

[19] Joachim Baumeister, Jochen Reutelshoefer, and Frank Puppe. Markups for knowledge wikis. In Handschuh et al. [63].

[20] Jos Kahan, Marja-Riitta Koivunen, Eric Prud'hommeaux, and Ralph R. Swick. Annotea: an open RDF infrastructure for shared web annotations. *Computer Networks*, 39(5):589–608, August 2002. doi:10.1016/S1389-1286(02)00220-7.

[21] Borislav Popov, Atanas Kiryakov, Damyan Ognyanoff, Dimitar Manov, and Angel Kirilov. KIM - a semantic platform for information extraction and retrieval. *Journal of Natural Language Engineering*, 10(3-4):375–392, September 2004.

[22] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[23] Atanas Kiryakov, Damyan Ognyanov, and Dimitar Manov. OWLIM – a pragmatic semantic repository for owl. In *Int. Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2005), WISE 2005*. Springer-Verlag LNCS series, November 2005.

[24] Sesame. `http://openrdf.org`. (last accessed on 26-11-2011).

[25] Christian Halaschek-Wiener, Jennifer Golbeck, Andrew Schain, Michael Grove, Bijan Parsia, and Jim Hendler. Annotation and provenance tracking in semantic web photo libraries. In Luc Moreau and Ian Foster, editors, *Provenance and Annotation of Data*, volume 4145 of *Lecture Notes in Computer Science*, pages 82–89. Springer, 2006. doi:10.1007/11890850_10.

[26] Jérôme Euzenat, Onyeari Mbanefo, and Arun Sharma. Sharing resources through ontology alignment in a semantic peer-to-peer system. In Yannis Kalfoglou, editor, *Cases on semantic interoperability for information systems integration: practice and applications*, pages 107–126. IGI Global, 2009. doi:10.4018/978-1-60566-894-9.ch006.

[27] Rohit Khare and Tantek Çelik. Microformats: a pragmatic path to the semantic web. In *Proceedings of the 15th international conference on World Wide Web*, pages 865–866. ACM, 2006.

[28] Maria Maleshkova, Jacek Kopeckỳ, and Carlos Pedrinaci. Adapting SAWSDL for semantic annotations of restful services. In *On the Move to Meaningful Internet Systems: OTM 2009 Workshops*, pages 917–926. Springer, 2009.

[29] Dan Brickley and Libby Miller. FOAF vocabulary specification 0.98. Technical report, xmlns.com, 2010.

[30] Scott Golder and Bernardo A. Huberman. The structure of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, April 2006.

[31] George A. Miller. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. doi:10.1145/219717.219748.

[32] Gail Hodge. *Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files*. Council on Library and Information Resources, April 2000.

[33] Lars M. Garshol. Metadata? thesauri? taxonomies? topic maps! making sense of it all. *Journal of Information Science*, 30(4):378–391, 2004.

[34] Barry Smith and Christopher Welty. FOIS introduction: Ontology–towards a new synthesis. In *FOIS'01: Proceedings of the international conference on Formal Ontology in Information Systems*. ACM Press, 2001. doi:10.1145/505168.505201.

[35] Fausto Giunchiglia and Ilya Zaihrayeu. Lightweight ontologies. In Ling Liu and M. Tamer Ozsu, editors, *Encyclopedia of Database Systems*. Springer, July 2009.

[36] Josef Kolbitsch. WordFlickr: a solution to the vocabulary problem in social tagging systems. In *Proceedings of I-MEDIA*, 2007.

[37] Pierre Andrews, Juan Pane, and Ilya Zaihrayeu. Semantic disambiguation in folksonomy: a case study. In *Advanced Language Technologies for Digital Libraries*, Lecture Notes on Computer Science (LNCS) Hot Topic subline. Springer, 2011.

[38] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer, 2007. doi:10.1007/978-3-540-76298-0_52.

[39] Simone Braun, Claudiu Schora, and Valentin Zacharias. Semantics to the bookmarks: A review of social semantic bookmarking systems. In *International Conference on Semantic Systems (I-SEMANTICS 2009), Graz, Austria*, pages 445–454, 2009.

[40] ISO. *ISO 2788:1986, Guidelines for the establishment and development of monolingual thesauri*. American National Standards Institute (ANSI), August 2007.

[41] Alistair Miles and José R. Pérez-Agüera. SKOS: Simple knowledge organisation for the web. *Cataloging & Classification Quarterly*, 43(3):69–83, 2007. doi:10.1300/J104v43n03_04.

[42] Andrew D. Miller and W. Keith Edwards. Give and take: a study of consumer photo-sharing culture and practice. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07, pages 347–356. ACM, 2007. doi:10.1145/1240624.1240682.

[43] Simone Braun, Andreas Schmidt, Andreas Walter, Gabor

Nagypal, and Valentin Zacharias. Ontology maturing: a collaborative web 2.0 approach to ontology engineering. In *Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge (CKC 2007) at the 16th International World Wide Web Conference (WWW2007)*, 2007.

[44] Michael Buckland. Vocabulary as a central concept in information science. In *Digital Libraries: Interdisciplinary Concepts, Challenges, and Opportunities. Proceedings of the Third International Conference on Conceptions of Library and Information Science*, May 1999.

[45] Adam Mathes. Folksonomies - cooperative classification and communication through shared metadata. Technical report, Graduate School of Library and Information Science. University of Illinois Urbana-Champaign, December 2004.

[46] Thomas Vander Wal. Folksonomy: Coinage and definition. `http://www.vanderwal.net/folksonomy.html`. (last accessed on 26-11-2011).

[47] Francesco Ronzano, Andrea Marchetti, and Maurizio Tesconi. Tagpedia: a semantic reference to describe and search for web resources. In *SWKM 2008: Intl. Workshop on Social Web and Knowledge Management @ WWW*, 2008.

[48] Karl Aberer, Philippe Cudré-Mauroux, Aris M. Ouksel, Tiziana Catarci, Mohand-Said Hacid, Arantza Illarramendi, Vipul Kashyap, Massimo Mecella, Eduardo Mena, Erich J. Neuhold, Olga De Troyer, Thomas Risse, Monica Scannapieco, Fèlix Saltor, Luca De Santis, Stefano Spaccapietra, Steffen Staab, and Rudi Studer. Emergent semantics principles and issues. In Yoon-Joon Lee, Jianzhong Li, Kyu-Young Whang, and Doheon Lee, editors, *Proceedings of the 9th International Conference on Database Systems for Advanced Applications (DASFAA'04)*, volume 2973 of *Lecture Notes in Computer Science*, pages 25–38. Springer, 2004.

[49] Martin Ames and Mor Naaman. Why we tag: motivations for annotation in mobile and online media. In *CHI 2007 - Tags, Tagging and Notetaking*, pages 971–980, 2007. doi:1240624.1240772.

[50] Sanjay C. Sood, Kristian J. Hammond, Sara H. Owsley, and Larry Birnbaum. Tagassist: Automatic tag suggestion for blog posts. *International Conference on Weblogs and Social Media*, 2007.

[51] Tye Rattenbury, Nathaniel Good, and Mor Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *SIRIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 103–110. ACM Press, 2007.

[52] Patrick Schmitz. Inducing ontology from flickr tags. In *Proc. of the Collaborative Web Tagging Workshop (WWW'06)*, May 2006.

[53] Aniket Kittur, Bongwon Suh, Bryan A. Pendleton, and Ed H. Chi. He says, she says: conflict and coordination in wikipedia. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07, pages 453–462. ACM, 2007. doi:10.1145/1240624.1240698.

[54] Rich Gazan. Social annotations in digital library collections. *D-Lib*, Volume 14 Number 11/12, December 2008.

[55] Manuele Quintarelli, Andrea Resmini, and Luca Rosati. Facetag: Integrating bottom-up and top-down classification in a social tagging system. In *IASummit*, 2007.

[56] Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The alignment API 4.0. *Semantic Web*, 2(1):3–10, 2011. doi:10.3233/SW-2011-0028.

[57] Siegfried Handschuh and Steffen Staab. *Annotation for the semantic web*, volume 96. IOS Press, 2003.

[58] Yolanda Gil, Mark Musen, and Jude Shavlik, editors. *K-CAP '01: Proceedings of the 1st international conference on Knowledge capture*. ACM, 2001.

[59] Siegfried Handschuh, Nigel Collier, Tudor Groza, Rose Dieng, Michael Sintek, and Anita de Waard, editors. *Proceedings of the Semantic Authoring, Annotation and Knowledge Markup Workshop (SAAKM2007) located at the 4th International Conference on Knowledge Capture (KCap 2007), Whistler, British Columbia, Canada, October 28-31, 2007*, volume 289 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[60] James Surowiecki. *The wisdom of crowds*. Anchor Books, 2005.

[61] Pierre Andrews, Juan Pane, Ilya Zaihrayeu, and Aliaksandr Autayeu. Supporting semantic annotations in flickr. In *IEEE International Conference on Intelligent Computer Communication and Processing*, 2011.

[62] Tobias Bürger, Ilya Zaihrayeu, Pierre Andrews, Denys Babenko, Juan Pane, and Borislav Popov. Report on the state-of-the-art and requirements for annotation representation models. Technical report, INSEMTIVES, June 2009.

[63] Siegfried Handschuh, Nigel Collier, Tudor Groza, Rose Dieng, Michael Sintek, and Anita de Waard, editors. *Proceedings of the Semantic Authoring, Annotation and Knowledge Markup Workshop (SAAKM2007) located at the 4th International Conference on Knowledge Capture (KCap 2007), Whistler, British Columbia, Canada, October 28-31, 2007*, volume 289 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.