# What The User Interacts With: Reflections On Conceptual Models For Semantic Wikis

François Bry, Michael Eckert, Jakub Kotowski and Klara Weiand

Institute for Informatics, University of Munich
Oettingenstr. 67, 80538 München, Germany
http://pms.ifi.lmu.de

**Abstract.** Traditional wikis excel in collaborative work on emerging content and structure. Semantic Wikis go further by allowing users to expose knowledge in ways suitable for machine processing, e.g. using Semantic Web technologies. The combination of ease of use, support for work in progress and Semantic Web technologies makes Semantic Wikis particularly interesting for knowledge-intensive work areas such as project management and software development. While several Semantic Wikis have been put to practical use, the concepts their users interact with have been little discussed. This position paper explores this issue, showing that the design of a conceptual model is not trivial and showing the repercussions of each design choice. The issue is explored stressing the social aspect of Semantic Wikis.

**Key words:** Semantic Wiki, Social Software, Semantic Web, Reasoning, Querying

## 1 Introduction

"Semantic Wiki" can refer either to Wikis enhanced with semantic technologies (or, after [8] "Semantic data for Wikis") or Wikis for ontology engineers (after [8] "Wiki for semantic data"). This paper uses the term in the first sense even though its contribution may be useful in both contexts.

Traditional Wikis are popular for managing personal and professional knowledge, primarily in the form of relatively simple hypertext, that is, Wiki pages and the links between them. Several features of traditional Wikis have been the key to their success: simplicity, openness and their thorough support for emerging and changing content in collaborative environments. Typically, Wikis are easy to use and allow anyone to make changes to the content. All Wiki content is version-controlled, meaning that previous versions of Wiki pages are never lost and changes can be tracked and reverted. Though many Wikis support advanced concepts that are relevant to their administration such as access rights and page and user groups, the basic concepts a regular users interacts with are limited to pages, links, and possibly text structuring and formatting.

Content in traditional Wikis consists of natural language text (and possibly multimedia files) and is not directly accessible to automated semantic processing. Therefore, knowledge in Wikis can be located only through simple user-generated structures (tables of contents, inter-page links) and simple full text keyword search. More advanced functionalities that are highly desirable in knowledge-intensive professional contexts such as querying, reasoning and semantic browsing are not possible. Semantic Wikis introduce capabilities into Wikis for specifying knowledge not just in natural language but also in more formal, machine-processable ways.

Most of the advanced technologies that Semantic Wikis employ were developed for use in a static environment with annotations and rules being crafted by knowledge representation experts. This is in contraposition to the ever-changing, dynamic character of Wikis where content and annotations are, for the most part, created by regular users. In such an environment, inconsistencies and ambiguities can easily arise and the system should therefore be able to cope with them and support users in their work.

While several Semantic Wikis have been put to practical use [1,7,17,16,5,12,21] [14], each using their own conceptual model, there has been little explicit theoretical exploration on the possible choices for conceptual models and their consequences [20]. By conceptual model, we here understand the basic concepts or buildings blocks that a user interacts with as well as how these building blocks relate to each other[1]. In a traditional Wiki, there are typically only few such building blocks, the most basic ones being "page" and "link." Semantic Wikis, however, add new building blocks such as typed links, tags and RDF or OWL annotations. The basic building blocks of a Semantic Wiki and how they relate to each other has rarely been discussed in the literature, and one can assume that many decisions in this regard have been without full consideration of the design space.

In this article, we seek to draw attention to this issue, showing that the design of a concept model for a Semantic Wiki is a non-trivial issue and design choices greatly influence how the user sees the system and what functionalities the system can offer. We will show that there are several possibilities for approaching certain issues in a Semantic Wiki and that these have advantages and disadvantages, as well as important consequences on how other issues can be approached.

## 2 Content

This section outlines the representation of content in the Wiki. "Content" here refers to text and multimedia which is used for sharing information, most frequently through the use of natural language, between the users of the Wiki, and whose meaning is not directly accessible for automatic processing. Information Extraction techniques can be used to extract structured data from text or speech, which enables computerised processing, but this introduces another level of representation which is not considered "content" in this sense.

[1] Since the term "concept" is overloaded, we refer instead to "building blocks" of a conceptual model.

While regular Wikis are restricted to content as data, Semantic Wikis add further layers, namely data that can be used for human as well as automatic processing or data that is intended only for computers and not easily understandable for humans. These two other types of data are discussed in the following two sections.

*Content Items.* Content items constitute the primary unit of information in the Wiki; a simple textual content item can be thought of as being similar to a paragraph or section in a formatted text. Content items give structure to Wiki content. A content item can directly contain only one type of content, for example text or video. However, content items are also compositional to enable the representation of complex composite content structure. Therefore, content items can be nested, yielding complex content items. Content items do not overlap and every content item has a URI and can be addressed and accessed individually. As a consequence, there is no inherent distinction between Wiki pages and content items, or rather, by default, all content items are Wiki pages. If every (simple or complex) content item can only be embedded in one other content item, the Wiki content consists of a set of finite trees. Root nodes, that is, content items that do not have a parent node, then have a special status in that they encompass all content that forms a cohesive unit. In this, they can be seen as being alike to a Wiki page in a regular Wiki.

Having an explicit concept of content structure in a Wiki is desirable both with respect to the semantic as well as the social nature of a Semantic Wiki as the structural semantics of the content can be immediately used for querying and reasoning as well as for facilitating collaboration and planning of content. For example, queries could be used to automatically generate tables of contents and the modular nature of content items facilitates collaboration and planning. In addition, content items constitute a natural unit for assigning annotations for content (see Section 3).

Allowing one content item to have several parents, that is, to be directly contained in multiple other content items through transclusion [9], is a design decision that adds functionality but also has side-effects, some of which may be unwanted. Allowing transclusion means that content items can be easily reused and shared, which is useful for example for schedules or contact data. If a copy of the content item's content is embedded, multiple occurrences of the content item in the Wiki can not be traced as naturally or easily. On the other hand, updating the content item or reverting it to an earlier version can lead to unintuitive and undesired effects as the content item changes in all contexts it is embedded in. The user editing the content item then needs to be aware of all the contexts in which the content item is used and has to ensure that the change to the content item is appropriate in all contexts. To facilitate this, information about embedding locations should be readily available to the users, but even then, the user is burdened with deciding whether the change can be made, something which he might not be willing or feel knowledgeable enough to do. When the transclusion of content items is enabled, loops, which arise when a content item contains itself as a descendant, pose another problem. This is due to the fact the resulting infinite recursion is problematic with respect to rendering the content item[2] as well as reasoning or querying over it. Since loops additionally appear to have no straightforward meaningful interpretation in the Wiki context, transclusions which would cause loops should generally be forbidden. In summary, allowing both content items that can be multiply embedded as well as content items that can only exist in one context combines the advantages of both strategies and gives the users maximum flexibility.

*Fragments.* Fragments are small, continuous portions of text (or, potentially, multimedia) that can be annotated with tags (see Section 3). While content items allow the authors to create and organise their documents in a modular and structured way, the idea behind fragments is that they constitute a means for users to annotate and use them separately from the original structure as they see fit and find useful. If content items are like chapters and sections in a book, then fragments can be seen as passages that readers mark; they are linear and in that transcend the structure of the document, spanning across paragraphs or sections and different sections of the book might be marked depending on which aspect or topics a reader is interested in.

Fragments should be maximally flexible in their placement, size and behaviour to allow for different groupings. Towards this goal, it is generally desirable that –unlike content items– fragments can overlap. The intersection between two overlapping fragments then can be further processed or it can be ignored. When two overlapping fragments $f_1$ and $f_2$ are tagged with "a" and "b" respectively, a third fragment that spans over the overlapped region and is tagged "a, b" could be derived automatically. Similarly, automatically taking the union of identically tagged overlapping or bordering fragments might be intuitive and expected by the user. However, this automatic treatment of fragments is a complex issue which might not always be appropriate or wanted.

On the other hand, fragments could be seen as co-existing but not interacting, meaning that the relationships between fragments are not automatically computed and no tags are added. This view has the advantage of being simpler and more flexible in that control of the fragments and their tags stays with the user. It is also in tune with the philosophy that, unlike content items that always only realise one structuring, fragments are individual in that different users can group a text in many different ways and under many different aspects. Fragments can either be restricted to the content directly contained in one content item, or it can span across content items. In the latter case, a rearrangement of content items can lead to fragments that span over multiple content items which no longer occur in successive order in the Wiki and, similarly, transclusion means that content items may contain only part of a fragment with the other part being absent (but present in some other content in which the content item is used).

Fragments could be deleted when the structure of content items no longer supports them, this means that a user might find a fragment she created destroyed as a consequence of another user's rearrangement of content items.

[2] At least if we assume that all of the content item is to be rendered at once.

Two possibilities of realising fragments are the insertion of markers in the text to label the beginning and end of an fragments ("intrusive"), or external referencing of certain parts of a content items, using for example XQuery, XPath, XPointer, or line numbers ("non-intrusive"). The latter has the advantage of allowing to define and annotate fragments on external content items, while the former means that fragments are less volatile and updates to the text do not affect fragments as easily, for example when text is added before the fragment.

*External Content Items.* Linked websites that are located outside of the Wiki are considered to be external content items. That means, they can be tagged and they can contain non-intrusive fragments, but they are not considered to be complex, that is, nested.

## 3 Semi-formal Annotations

One problem that frequently arises in the context of Semantic web applications is that it is hard to motivate users to annotate content since they find the process complicated and laborious. One solution is to provide means for creating less formal annotations which are easier to use. As work progresses, these annotations can be made increasingly more precise and can eventually be transformed into formal knowledge. Tagging is one such kind of semi-formal annotation. Tags normally consist only of keywords users associate with resources. Despite their simplicity, there are many possibilities as to how exactly the tags should work and be used [19]. Further, traditional keyword tagging can be extended in a number of ways [2,23,18] such as structured tags, negative tagging, and rules for tags [6]. Semantic links are another kind of semi-formal annotation. They are anchored in content items or fragments and can point to content items or fragments. Tags can be used on content items, fragments and possibly links as a way to assign a type to a link.

*The semiotic triangle.* One question to ask when designing a system that includes annotations is "What is annotated?" This question may have a quick, superficial answer: "Any resource that the system allows to be annotated." But what does that mean precisely? Let us say that the resource is a Wiki page about an elephant. Does a tag added to the page state a fact about the page itself (a representation of an elephant in the Wiki system) or does it refer to the actual elephant? This leads to a concept known as semiotic triangle [10], Peirce's triad [13] or de Saussure's distinction between the signifier and the signified [15]. This distinction is important because it has consequences on how the annotations are interpreted. In [11], the authors let the users decide what exactly they want to express by providing them with a syntax that allows the users to distinguish between these two cases.

Although it may not be important for the user, for the system design, it is essential to differentiate tags from tag associations (or "taggings"). Users connect tags to resources which is reflected in the system by the creation of a tag association, which, apart from the user, the tag and the tagged resource, may involve additional information such as the time of the tagging event.

*Structured tags.* Ordinary flat tags are limited in their expressiveness. To overcome this limitation, different extensions of tagging are currently being proposed: machine tags[3], sub-tags [2] as used in the website http://www.rawsugar.com/, structured tags [2], etc. Most of the proposals are a variation of keyword:value pairs, some extend it to full RDF triples [23]. Note that keyword:value pairs can be seen as triples, too - the resource being annotated is the subject, the keyword is the predicate and the value is the object of the triple. More complex schemes which involve nesting of elements might be practical in some cases, e.g. "hotel(stars(3))" could express that the tagged resource is a three-star hotel. These extensions develop the structure of the tag itself and a set of tags is interpreted as a conjunction. It is conceivable to allow users to tag resources with a disjunction of tags or even with arbitrary formulae. This may be practical for some applications but it has two drawbacks: 1) reasoning with disjunctive information is difficult, 2) simplicity and intuitiveness would suffer.

*Negative tags.* So far, we have only addressed expressing positive information. In a collaborative context, we may be interested in tracking disagreements which presupposes some way to express negative information, such as negative tagging. If the user is allowed to tag a resource with tag "t" he or she may want to tag it with "not t" as a way to express disagreement or to simply state that the resource is not "t" or does not have the property "t". An example may be a medical doctor tagging a patient's card as "not lupus" to state that the patient definitely does not have "lupus". There are two ways to interpret negative tagging. It might be seen as classical negation or it may be seen as a kind of voting to express agreements and disagreements (see Section 5). Although a tag "not t" could be seen as introducing classical negation into the system, it may in fact be only a very weak form of negation because we can allow negating only pure tags, not general formulae (or sets of tags), and the only way to interpret this kind of negation would be by introducing a rule which says that from tag "t" and tag "not t" a contradiction symbol shall be derived (for more about reasoning see Section 6).

*Tags as concepts.* A hindrance in the transition from tags to more formal knowledge (e.g., RDF triples) is that tags are just keywords (i.e., strings). Often different keywords can be used to express the same abstract concept (e.g., keywords in different languages, synonyms). Similarly, the same keyword might be used to express different concepts (e.g., homonyms like "bank"). A possibility that fits well in the Wiki context, is to separate concrete keywords and the abstract concepts by using content items (which represent the abstract concepts) instead of keywords for tagging. Keywords still play an important role, as they are what is entered by the user, but the system will automatically resolve them to corresponding content items, possibly interactively asking for clarifications in the

[3] http://tech.groups.yahoo.com/group/yws-flickr/message/2736

case of ambiguities. In systems supporting semantic browsing over tags, there is also a natural need to have (partly automatically generated) content items for tags, e.g., to provide a list of all content items being tagged with a particular tag.

Using content items as tags also solves some further issues beyond synonyms and homonyms. Unlike keywords, content items have a URI that can be used when transforming information of semi-formal tags into formal RDF models (e.g., by the use of rules). Even more importantly, content items also offer a place for describing tags further. This encompasses both natural-language explanations for humans on the meaning and intended use of the tag as well as machine-readable descriptions, e.g., by means of tagging a tag's content item (see also tag hierarchies further down).

*Links.* Links are primarily used for navigation but can also be considered a kind of annotation. With respect to annotation, links can be seen as a way to specify some kind of relation between the two linked resources. For an untyped link, this relation may default to the "is related to" relation. Typed links express a specific relation between two resources. Link type is a new concept in the usage model which could be unified with the rest of the system by letting the user specify the type by tagging the link. Advantages of this approach are the intuitiveness of tagging and the social, work-in-progress aspect of the environment which allows the users to converge on a precise meaning of the link only as their work progresses (e.g. by discussing the link type on the page of the tag). Disadvantages are unclear meaning of a link with multiple tags and possible user interface issues. A question that arises with links with multiple tags is how they are interpreted for reasoning, querying, and translation to formal knowledge (e.g. RDF). For this consider a link between resources $R_1$ and $R_2$ with tags A and B. Can it be distinguished from two links between resources $R_1$ and $R_2$, one with tag A, the other one with tag B? Treating multiple tags as multiple links, i.e. not distinguishing the two situations, is simpler for translation to RDF because then each link maps directly to one triple where tags correspond to properties. If on the other hand they are to be distinguished, a new property has to be introduced to express that the link is tagged with both A and B.

*Tags vs links.* When tags as concepts are supported, simple flat tags on content items can be seen as a kind of link between the tagged resource and the concept of a tag represented by the content item describing its meaning (see above). Similarly, structured tags, such as keyword-value pairs, can be seen as expressing a relation (or a link), with its type given by the keyword, between the tagged resource and another resource, given by the value. In a Wiki supporting semantic browsing over such tags, the question may then arise of what differentiates a link from tags and structured tags. From a technical point of view there may not be a strict differentiation after all, flat tags can be seen as specialised links between a taggable resource and the content item describing the concept of the tag, as a link is then a general way of expressing a relation. The difference usually is in the way they are presented and used. Tags are usually represented separately from a content item, e.g. in a special area of the page, while links are represented with anchors inside the content item. Further, tags make a statement about a single content item, e.g. give it a type, whereas purpose of links is to express an association between two content items. Finally, while links can be tagged, one cannot link to or from another link.

*Tag Hierarchies.* Tag hierarchies constitute a step in the transition from informal to formal annotation. They are useful for example for reasoning and querying since they enable the processing of tag relationships. Tag hierarchies could be created through "tagging tags", that is, tagging a tag's content item to indicate an "is-a" relation.

Semi-formal annotations described in this section provide a means to transform knowledge from human-only content described in Section 2 to machine-processable information described in the next section. Semi-formal annotations seem to be an important feature of social software because they provide a low-barrier entry point for user participation on enrichment of content with metadata which are machine processable. Users can use gradually more expressive and formal methods of annotation as they become familiar with the system. First, they only create and edit content. Then they can begin using flat tags to annotate content and later perhaps start using structured tags. Advanced users or system administrators can further enhance the metadata enrichment efforts by specifying rules for semi-formal annotations, see Section 6.

## 4 Formal Knowledge Representation

Currently, the most common format for semantic data is RDF. RDF data are easily processable by machines but not easily interpretable by Wiki users. They can use semi-formal annotations which can then be represented directly in RDF or be later transformed to formal annotations that use vocabularies with well-defined meanings. This transformation can be supported by rules or by methods that automatically extract folksonomies from sets of tags. In this approach, ontologies could naturally emerge based on on-demand basis as a formalisation of semi-formal annotations of Wiki content. Semi-formal annotations such as flat tags, structured tags and links can be easily translated into RDF triples, meaning that RDF is a suitable choice for the representation of all annotations. It may be the case for the low-level implementation of the system but it is not desirable to let the user write or read raw RDF data. It is usually obvious how to use a tag but it is not obvious how to write an RDF triple. Therefore the user should be rather exposed to higher-level, intuitive concepts such as tag, structured tag and link which can then be automatically translated into RDF. For practical applications, support of RDF is important also because of interoperability with current semantic web applications and linked data [4,3]. A social Semantic Wiki should therefore support at least import and export of RDF data.

## 5 Social Content Management

To facilitate social collaboration and leverage the social aspects of the Semantic Wiki, several options and aspects have to be considered.

*Groups.* User groups can be used among other things for personalisation of wiki content, for querying and reasoning and to attribute wiki data to a group. Tags are an easy way to group things which is used in the wiki, so it would be an obvious choice to form user groups by tagging users' content items. Every tag that is used on at least one (or possibly two) user's content item then constitutes a group. One possible drawback of this approach is that this proliferation of groups might be demanding in processing when special mechanisms for treating groups are established.

*The social weight of tags.* If several users tag one item with the same tag, it is natural to aggregate these tag assignments to give a clearer view of all the tags assigned. Tags then can be seen to have weights depending on how often they were assigned. On the other hand, other users might not agree with the assignment of a certain tag to a content item, adding a negative component to the tags' weight. The overall social weight of a tag can then be calculated for example by weighting the number of positive assignments versus the sum of both positive and disagreeing assignment, or by assigning a value to both actions and calculating the total. The social weight of a tag then summarizes the users' views on the appropriateness of a specific tag assignment and thus provides a valuable measure that can be used in reasoning and querying. Moreover, reinforcing or disagreeing about tag assignments constitutes a low-barrier activity in the wiki, making it easy for beginning users to participate.

Agreement or disagreement could also be expressed with respect to a content item itself, for example through a specific set of tags which are reserved for this use.

*Access Rights.* Users, user groups and rules for reasoning could be used to handle access rights in the wiki, but this is a complex issue which requires further investigation. Questions that arise include who owns the rules and what are the access rights on rules and who can assign the tags that restrict the access. Static rules would not be suited for rights managements in all environments. For them to function well, the organization and roles in the wiki have to be relatively stable, which may be the case in professional applications. In other areas, such as the development of open source software, such rules may not be desired or the social organization might not be static enough for rules to be adequate.

## 6 Reasoning

Reasoning is enabled by the formal and semi-formal annotations in Semantic Wikis. Wiki content can change frequently, disagreements are common and inconsistencies and ambiguities can easily arise. This is not only unavoidable but even desirable in a creative environment and reasoning should be able to cope with it and support users in their work-in-progress. Reasoning that has these properties was sketched previously in [6]. As indicated in previous sections, it is also desirable that social software supports a rule language for annotations that would help users to further annotate content. A rule might for example express that a tag "elephant" induces an implicit "mammal" tag. Another example might be a rule expressing that a tag "bug report" without a corresponding tag "processed" induces an implicit "todo" tag.

The second example of a rule presupposes that the rule language includes negation as failure. This choice of negation seems to be appropriate for a Wiki. A Wiki is, in a way, a world of its own. For example, if there is a page describing a meeting with a list of tasks to be done then it is safe to assume that the list is complete, i.e. there are no other tasks for this meeting other than the ones listed on the page. Also, if a meeting is not tagged as "all-hands" then it is safe to assume that it is not an all-hands meeting. Therefore negation as failure seems to be the negation of choice for a Semantic Wiki. On the other hand, negative tagging, as discussed in Section 3, expresses negative information explicitly. Therefore the user could express not only positive and negative information but also refer to missing positive and missing negative information. The problem of combining classical negation with negation as failure is a field of its own and it cannot be expected that a regular user would understand it. On the other hand, recall that, in our approach, negative tags can be just positive tags with a negative marker that is interpreted only using a rule "from tag t and tag not t derive a contradiction *symbol*". This results only in a very weak form of negation that should not be too difficult to combine with negation as failure in an intuitive way. The derivation of a contradiction symbol rather than a contradiction enables paraconsistent reasoning, see [6] for details. The reasoning approach sketched in [6] has also a social aspect in that it allows to track different inconsistencies to their origin and thus can provide users with useful information about the cause of each inconsistency which may be a result of a disagreement within a group of people or simply a mistake.

It may be interesting and beneficial for users to see how a specific group of users tagged a resource or to compute the ratio of the number of people agreeing and disagreeing with an item. Therefore the rule language should include aggregation and be sufficiently expressive to allow referring to tags by different users associated to different content items.

## 7 Querying

A query language for a Semantic Wiki should enable users to select, access and reuse data while leveraging the Wiki's properties to improve retrieval, ranking and the "mashing-up"/embedding of existing data. Traditional Wikis frequently use full text search, while several query languages have been proposed for semantic web data, specifically XML and RDF [22]. Full text search is simple but not powerful. Semantic web query languages on the other hand, are too complicated for casual users, although some efforts to enable keyword-based querying of RDF and XML. Furthermore, relatively little research has been made in the area of combining querying of textual data and annotations.

However, in the Semantic Wiki, all conceptual building blocks, for example content items and their structure, tags and tag hierarchies should be amenable to querying and it should be possible to combine selection criteria for several data sources in one query, for example expressed as label:keyword terms where the label specifies a datasource (text, tags) or property (bold, author, time added) and the keyword is a string.

Further, the query language needs to be versatile and user-friendly, meaning that users should be able to enter just a keyword and get meaningful results, while more experienced users should be able to specify complex queries. The transition between the two needs to be smooth and flexible. A suitable query language for a Semantic Wiki should allow for aggregation and construction of results to create views in the form of content items composed of Wiki data. Similarly, queries embedded in content items may be used to always display up-to-date query results that change as the Wiki content does.

Finally, the ranking of results could utilise properties specific to the Semantic Wiki like tag weights, edit frequency, the number of hits or the author's expertise or equity value to improve ranking results.

## 8 Conclusion

In this article, we explored the conceptual bulding blocks of Semantic Wikis and outlined choices that have to be made when designing a Wiki with advanced functionalities. There are many options and details we could not discuss here for space reasons. One vital question we had to omit is how the two kinds of data – content and annotations– are to be handled with respect to versioning, such as whether content and annotations should be versioned together or separately. We also did not discuss a complex method of measuring social weight of tags and content items called community equity. Community equity is employed by Sun Microsystems, a partner in the KiWi project, in their internal portal SunSpace and is used to encourage user participation by showing them the importance of their contributions to the community (i.e. community equity). This paper focused on two of the advanced wiki functionalities, namely reasoning and querying. There are other advanced functionalities such as personalisation and information extraction which affect the design decisions as well. For example fragments are an important concept for annotation by means of information extraction and rules and groups play an important role in personalisation. Also, many Semantic Wikis have already been implemented and it would be worthwhile survey the design decisions that were made in these existing systems; to the best of the authors' knowledge no such survey exists yet[4].

[4] Although recently, there has been a related effort to create a "Semantic Wiki Feature Matrix", see http://semanticweb.org/wiki/Semantic_Wiki_State_Of_The_Art

*Acknowledgements* The research leading to these results is part of the project "KiWi - Knowledge in a Wiki" and has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 211932.

## References

1. S. Auer, S. Dietzold, and T. Riechert. Ontowiki-a tool for social, semantic collaboration. *International Semantic Web Conference*, 4273:736–749, 2006.
2. Judit Bar-Ilan, Snunith Shoham, Asher Idan, Yitzchak Miller, and Aviv Shachak. Structured vs. unstructured tagging a case study. 2006.
3. T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets. Tabulator: Exploring and Analyzing linked data on the Semantic Web. In *Proceedings of the 3rd International Semantic Web User Interaction Workshop*, volume 2006, 2006.
4. Tim Berners-Lee. Linked data. *W3C Design Issues*, 2006.
5. Philip Richard Boulain. Swiki: A semantic wiki wiki web. Master's thesis, University of Southampton, 2005.
6. François Bry and Jakub Kotowski. Towards reasoning and explanations for social tagging. *Proc. of ExaCt2008 - ECAI2008 Workshop on Explanation-aware Computing. Patras, Greece*, http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2008-2, 2008.
7. Kensaku Kawamoto, Yasuhiko Kitamura, and Yuri Tijerino. Kawawiki: A template-based semantic wiki where end and expert users collaborate. In *Proceedings of 5th International Semantic Web Conference (ISWC2006)*, 2006.
8. Markus Krtzsch, Sebastian Schaffert, and Denny Vrandecic. Reasoning in semantic wikis. In *Reasoning Web Summer School 2007*, pages 310–329, 2007.
9. T.H. Nelson. *Literary Machines*. Mindful Press, 1993.
10. C.K. Ogden, I.A. Richards, B. Malinowski, and F.G. Crookshank. *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*. Harcourt, Brace & Company, 1938.
11. E. Oren. Semantic Wikis for Knowledge Workers. 2005.
12. Eyal Oren. Semperwiki: a semantic personal wiki. In *Proceedings of 1st Workshop on The Semantic Desktop - Next Generation Personal Information Management and Collaboration Infrastructure*, Galway, Ireland, 2005.
13. C.S. Peirce. On a new list of categories. In *Proceedings of the American Academy of Arts and Sciences*, volume 7, pages 287–298, 1868.
14. Niko Popitsch, Bernhard Schandl, Arash Amiri, Stefan Leitich, and Wolfgang Jochum. Ylvi - multimedia-izing the semantic wiki. In *Proceedings of the 1st Workshop "SemWiki2006 - From Wiki to Semantics"*, Budva, Montenegro, 2006.
15. F. Saussure. Course in general linguistics (W. Baskin, Trans.). *New York: Philosophical Library*, 1916.
16. Sebastian Schaffert, François Bry, Joachim Baumeister, and Malte Kiesel. Semantic wikis. *IEEE*, page 7, 2008.
17. Sebastian Schaffert, Rupert Westenthaler, and Andreas Gruber. Ikewiki: A user-friendly semantic wiki. In *3rd European Semantic Web Conference (ESWC06)*, Budva, Montenegro, 2006.
18. B. Sereno, B. Shum, and E. Motta. Formalization, User Strategy and Interaction Design: Users' Behaviour with Discourse Tagging Semantics. 2007.

19. Gene Smith. *Tagging: People-powered Metadata for the Social Web (Voices That Matter)*. New Riders Press, December 2007.
20. R. Tolksdorf and E.P.B. Simperl. Towards wikis as semantic hypermedia. In *Proceedings of the 2006 international symposium on Wikis*. ACM New York, NY, USA, 2006.
21. Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller, and Rudi Studer. Semantic wikipedia. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 585–594, New York, NY, USA, 2006. ACM.
22. K. Weiand, T. Furche, and F. Bry. Quo Vadis, Web Queries? In *Proc. Int'l. Workshop on Semantic Web Technology (Web4Web) 2008*, 2008.
23. Jie Yang, Yutaka Matsuo, and Mitsuru Ishizuka. Triple tagging: Toward bridging folksonomy and semantic web. *ISWC07*, page 14, 2007.