SweetWiki: Semantic Web Enabled Technologies in Wiki

Michel Buffa
Mainline Group, I3S Laboratory, University of Nice
Acacia Group, INRIA Sophia-Antipolis
France
33 4 92 38 50 15
buffa@unice.fr

Fabien Gandon
Acacia Group, INRIA laboratory, Sophia-Antipolis, France
33 4 92 38 77 88
Fabien.Gandon@sophia.inria.fr

ABSTRACT
Wikis are social web sites enabling a potentially large number of participants to modify any page or create a new page using their web browser. As they grow, wikis may suffer from a number of problems (anarchical structure, aging navigation paths, etc.). We believe that semantic wikis can improve navigation and search. In SweetWiki we investigate the use of semantic web technologies to support and ease the lifecycle of the wiki. The very model of wikis was declaratively described: an OWL schema captures concepts such as wiki word, wiki page, forward and backward link, author, etc. This ontology is then exploited by an embedded semantic search engine (Corese). In addition, SweetWiki integrates a standard WYSIWYG editor (Kupu) that we extended to support semantic annotation following the "social tagging": when editing a page, the user can freely enter some keywords and an auto-completion mechanism proposes existing keywords by issuing queries to identify existing concepts with compatible labels. Thus tagging is both easy (keyword-like) and motivating (real time display of the number of related pages) and concepts are collected as in folksonomies. To maintain and reengineer the folksonomy, we reused a web-based editor available in the underlying semantic web server to edit semantic web ontologies and annotations. Unlike in other wikis, pages are stored directly in XHTML ready to be served and semantic annotations are embedded in the pages themselves using RDFa. If someone sends or copies a page, the annotations follow it, and if an application crawls the wiki site it can extract the metadata and reuse them. In this paper we motivate our approach and explain each one of these design choices.

Categories and Subject Descriptors
K.4.3 [**Organizational Impacts**]: Computer-supported collaborative work.

General Terms
Management, Measurement, Documentation, Experimentation, Human Factors, Standardization.

Keywords
Wiki, Semantic Web, Social Tagging, Ontology, Web 2.0.

1. INTRODUCTION
Why did wikis become such a phenomenon? At WikiSym 2005, Ward Cunningham and Jimmy Wales provided some elements of an answer: *"a wiki is like a garden; users (...) must take care of it. Start with some seeds and watch it grow, and the wiki will become moderated by its users' community, (...) respect and trust the users, (...) good things happen when you trust people more than you have reason to, let everybody express his opinion, no censorship, consensus must be reached, (...) the wiki is adapted to a dynamic social structure because of its refactoring features (...) Do not impose a rigid structure, users will refactor and structure the wiki as it grows (...)"* [18, 19]. This sounds revolutionary and indeed, social aspects are important and cannot be neglected when talking about wikis. Wikis introduced groundbreaking innovations as a technology supporting collaborative web authoring, but also at the level of the process, philosophy and even sociology of such collaborative authoring [16, 17]. However, even when wikis have been adopted by a large community, they may suffer from a number of problems. The main problem reported is the open structure that makes navigation, orientation and search difficult [2, 3, 23]; wikis often fail to scale with the number of pages.

Wikipedia defines a Semantic Wiki as a *"Wiki that has an underlying model of the knowledge described in its pages. (...). Semantic Wikis allow capturing or identifying further information about the pages (metadata) and their relations. Usually this knowledge model is available in a formal language, so that machines can (at least partially) process it".* We believe that semantic wikis can be searched, navigated and shared with other applications in better ways than regular wikis. SweetWiki is such a semantic wiki. To address the lack of structure and structuring tools SweetWiki integrates semantic web technologies at the core of its wiki engine. It does so without changing the ease of use that makes wikis so popular.

In section 2 we focus on the problems encountered by large wikis, in particular navigation and search, and we will explain the concepts of social tagging and folksonomies as means to improve navigation and search. In section 3 we present SweetWiki in details and insist on its innovative features. In section 4 we present related works and compare them to SweetWiki. Finally we discuss the future of semantic wikis and we mention the extensions we are working on.

1. MOTIVATING REMARKS
Few academic papers have addressed the intranet-wiki topic [1]. In [3] we detailed two experiences we conducted over several years with intranet wikis: (1) six years ago we installed a wiki which is today at the heart of the intranet of the Computer Science department of the University of Nice, with about 400 regular users [4]; and (2) since 2001, we have a close relationship with

the ILOG Company which has developed an impressive wiki-powered intranet [2].

Companies like Google, Motorola and the New-York Times have made public the way they use a wiki in their organization [5, 6, 7, chapter 12 of 8]. In [3] we defined the goals of a business organization intranet and showed how the web technology and tools helped or failed to reach these goals. We focused on the wiki concept and concluded that its success relies on several social conditions that cannot always be found in the business organization's culture (e.g. people must understand why they are working together; there must not be too much social friction, etc.)

Finally, wikis may suffer from a number of problems. The main problem reported is the difficulty experienced by users in finding their way, in navigating and searching the wiki, especially when it becomes large. Traditional wikis do not scale well unless their structure and structuring tools are improved.

ILOG uses the aspSeek search engine to index and search the resources of their wiki-based intranet. Looking at their logs over time it become apparent that the use of the search engine suddenly dropped, and after a short time people just stopped using it. Interviews and investigations proved that the assumption that everybody knows how to use a search engine was wrong [2, 3]. In addition, on the Internet, people can accept not finding what they are searching for -maybe it is just not out there; on an intranet, when people know that what they are looking for is there, they don't understand why a search engine does not find it and finally distrust it altogether. After many usability tests, the user interface for the search engine was improved, but people still complain about the difficulty to find things on the wiki. The interviews and questionnaires at the University of Nice confirmed the same problems with their wiki: search is considered less and less useful as the wiki grows [3].

The New York Times Digital used a wiki that became huge, with thousands of pages and several problems occurred [8]. They first added navigation bars, but this did not solve the navigation problem. WikiNames collision was another problem: when one creates a wiki page one has to choose a name for this page; after two years, users sometimes had to try dozens of different names before finding a name that had not already been used. The original idea with WikiNames collision was that if you find out that there is a page that already exists with the same name, you would "join it" because it is supposed to be the best place for saying what you have to say. But it just did not work at the NY Digital: people wanted to create their own page. They invented WikiNames that were no longer meaningful according to their content. Navigation and searching were so difficult that it was nearly impossible to find a document without having bookmarked it. Everybody realized that the wiki was becoming a mass of increasingly inaccessible pages but the user community was not ready to do the necessary work for refactoring and organizing it all. The writing and publishing process in a national newspaper is very structured, and NY Times Digital's employees could not get any trace of such a workflow in the wiki. What appeared as a promising tool that had been widely adopted turned out to be a problematic solution for helping in the publishing process. Structure and organization became such a big problem that they had to stop relying on a wiki. It was not completely abandoned but relegated to a shared notepad, with the structured work being done in other tools.

One can argue that the choice of another wiki engine could have changed the outcome of this experience, in particular a wiki engine supporting the concept of workspaces like TWiki, MoinMoin, JotSpot, SocialText, etc. But we think the problem runs deeper. Wikis are designed to be structured by the users themselves. People differ from each other, every individual has his own way of classifying and organizing data, and this may change over time. A hierarchical structure like the one proposed by the workspaces is certainly a good thing from a technical point of view but it provides a superficial modularization of a wiki [14]. Horizontal navigation (following links in the page itself) is the means most people use. Usability tests showed that most people at ILOG don't even know the names of the different workspaces.

Interestingly, a common behavior we noticed is that users started to add category keywords on the wiki pages. These keywords are WikiNames that lead to pages that propose hyperlinks to all pages belonging to the same category. This naïve classification helps but does not scale. We drew a parallel between this emergent behavior and the phenomenon of social tagging used in the public Web by popular sites such as del.icio.us and flickr.com and also widely used in blogs. You can annotate your blog entries or the pictures you posted to flickr by associating keywords to them forming a quasi-classification on-the-fly. These tags are used by technorati.com's web bots and a link to your tagged resource is added to the other entries that share the same tag. The main interest in this way of tagging is its social approach to classification. People can use whatever tags they feel represent the content of their writing, but they may find out that this tag has never been used before. So there is a higher probability they will add other tags that link them to other resources. If one creates a new tag, it is just added and will be proposed as a choice when another person enters a tag that starts with the same letters, and maybe this person will in turn choose it. This way, users as individuals, can categorize their writing any way they want and at the same time begin a grass roots taxonomy or *folksonomy*.

Social tagging and folksonomies are the subjects of debate in different communities, including the semantic web community [12]. These concepts are often described as an alternative to ontologies and to the semantic web approach in general [11, 15]. Gruber in [15] published an interesting survey of these different points of view. Some describe tags and folksonomies as "cheap metadata for the masses" (taxonomies and ontologies being the land of experts) [33] and others think they are the one true way [11] and that a flat-hierarchy is more human-brain-friendly, imitating the word-as-a-label-for-things. But this is also the main drawback of the tags: human-language-structured thought can jump between concepts; the same word can have totally different meanings. Last but not least: each human has his own world-experience, his own tagging-system that may not be generalized. Where categories are managed by specialists to achieve the best classification, tags are users' rough approximation of classification for a practical use (*ethnoclassification).*

2. SWEETWIKI

Wikis were designed in the mid-nineties exploiting the web technologies of the time *i.e.* mainly HTML, HTTP and URIs. To make up for the lack of simple remote edition and storage facilities Wikis developed WikiML variants (wiki markup languages), WikiWords for specifying hypertext links, simple versioning mechanisms, etc. The idea of SweetWiki is to revisit

the design rationale of Wikis, taking into account the wealth of new standards available for the web eleven years later to address some of the shortcomings identified through experience.
After evaluating several wiki engines (regular or semantic), we decided to write a new engine because our vision of the wiki of the future was not compatible with what we found in existing wikis. We wanted our wiki to:

- *rely on web standards*: standards for the wiki page format (XHTML), for the macros one can put in a page (JSPX/XML tags), etc.;
- *be articulated around a semantic engine* that supports semantic web languages like RDF, RDFS, OWL, SPARQL, etc.;
- *get rid of the WikiML* dialects used and modified by most wiki systems. We took this decision based on the painful experiences we had with the ILOG intranet during the introduction of WYSIWYG editors in TWiki. We encountered many problems during the translation between the WikiML and the XHTML languages. Many WikiML variants do not support all the XHTML produced by the existing editors. Mixing WikiML with XHTML code was not a clean approach and users were asking for more intuitive interfaces. Furthermore, we wanted an alternative to translating WikiML to XHTML each time a page is viewed and doing the reverse translation each time a page is saved.
- propose faceted navigation and enhanced search tools;
- propose metadata editing in the same user interface used for content editing.

2.1 Principles

Wikis are Web sites where pages are organized around WikiWords and sometime other constructs such as WikiWebs or Workspaces. To go beyond this informal hyperlink structure, semantic tagging and restructuring functionalities are needed. To make explicit, manipulate and exploit such a structure we introduced two ontologies:

- *an ontology of the wiki structure*: the wiki concepts are usually buried in their *ad hoc* implementations; this structure is a special kind of meta-data (forward links, authors, keywords, etc.) relying on an ontology of wikis (WikiPage, WikiWord, WikiWeb, etc.). By making this structure and its ontology explicit, we can reason on it (e.g. to generate navigation pages) we can modify it (e.g. re-engineer the wiki structure) and we can build on it (e.g. interoperability between several wikis).
- *an ontology of the topics*: each wiki page addresses one or more topics. In order to ease navigation while maintaining the usual simplicity, we implemented the usual tag/keyword mechanism with a domain ontology shared by the whole wiki. By making this topic ontology explicit we can once again reason on it (e.g. find semantically close topics) make complex queries (e.g. find pages tagged with close topics), we can modify it (e.g. tidy the ontology, merge equivalent concepts, etc.)

The ontology of the wiki structure is maintained by developers of the wiki. The domain ontology is enriched directly by the users and may be restructured by administrators or volunteers of the site to improve the navigation and querying capabilities. Other ontologies may be added at runtime and be immediately accessible to users. To implement these principles we relied on a semantic web server architecture described in the following section.

2.2 Architecture

Figure 2 summarizes the architecture of SweetWiki. The implementation relies on the CORESE semantic search engine for querying and reasoning [38] and on SEWESE, its associated web server extension that provides API and JSP tags to implement ontology-based interfaces, as well as a set of generic functionalities (security management, ontology editors, web application life cycle, etc.)
The server relies on a standard web application architecture: filters manage the session (e.g. authorization, user profiles, etc.) and the template of the site (headers, trailers); pages are directly available in XHTML or JSPX for browsing; a servlet handles saved pages; a set of JSP tags provide high level functionalities (e.g. submit a SPARQL query and format the result with an XSLT stylesheet); javascript libraries are served to provide a WYSIWYG editor (based on Kupu).
Starting from the users' side, SweetWiki is based on Kupu[34] an XHTML editor in JavaScript which allows us to replace traditional WikiML editing by a WYSIWYG interface in the user's browser. The directly produced XHTML is the persistence format. Thus, once saved, a page stands ready to be served by the Web server.
Pages are standalone XHTML files including their metadata and thus they can be crawled by other applications. To address structuring and navigation problems in wikis we wanted to include tagging at the core of the wiki concept, thus we integrated four new web technologies:

1. RDF/S and OWL are W3C recommendations to model metadata on the web [35];
2. SPARQL is a recommendation for a query language for RDF [39];
3. RDFa is a draft syntax for Embedding RDF in XHTML [36];
4. GRDDL is a mechanism for getting RDF data out of XML and XHTML documents using explicitly associated transformation algorithms, typically represented in XSLT [37].

The RDF model has an XML syntax but it is currently impossible to validate documents that contain arbitrary RDF/XML tags and therefore it is problem to import RDF/XML into other markup languages such as XHTML. On the other hand, the external annotation of document in RDF/XML can result in significant data duplication between the actual annotated resource and the RDF/XML annotation. For the sake of maintenance, concision, and encapsulation it is often better to add RDF to a document without repeating the document's existing data. RDFa proposes a solution to augment existing markup with metadata, using class and property types defined in RDF Schemas, combined with the existing content from the host language.
In XHTML using RDFa, a subject is indicated using the attribute `about` and predicates are represented using one of the attributes `property`, `rel`, or `rev`. Objects which are URI-referenced are represented using the attribute `href`, whilst objects that are literals are represented either with the attribute `content`, or the content of the element annotated.
The example in figure 1 shows how a tag `<a>` is augmented with the attribute `rel` to annotate a `blockquote` with the URL of its source according to the Dublin Core ontology.

```
<blockquote>
  As defined in <a rel="dc:source"
  href="http://en.wikipedia.org/wiki/Semantics">
  the wikipedia pages</a>, semantics is the study
  of meaning but it is distinguished from
ontology
  in being about the use of a word more than the
  nature of the entity referenced by the word.
</blockquote>
```

Fig 1. example of RDFa

Contrarily to external RDF annotations, this approach is inline with the wiki spirit where everything is done in the page: anything can be copied and pasted in one place (the wiki page) even using a WYSIWYG editor. With RDFa we have both page data and metadata in the same standalone file (XHTML) and pages can be crawled by external applications or saved by users using their browser without any loss of information.

When the servlet receives a page to save it also applies the GRDDL XSLT stylesheet to it in order to extract the RDF embedded in it and output it as RDF/XML files.

Any classical wiki query is done relying on the Corese engine. For instance when a new page is created corresponding to a WikiWord, to identify other pages referencing this WikiWord, the servlet issues a SPARQL query.

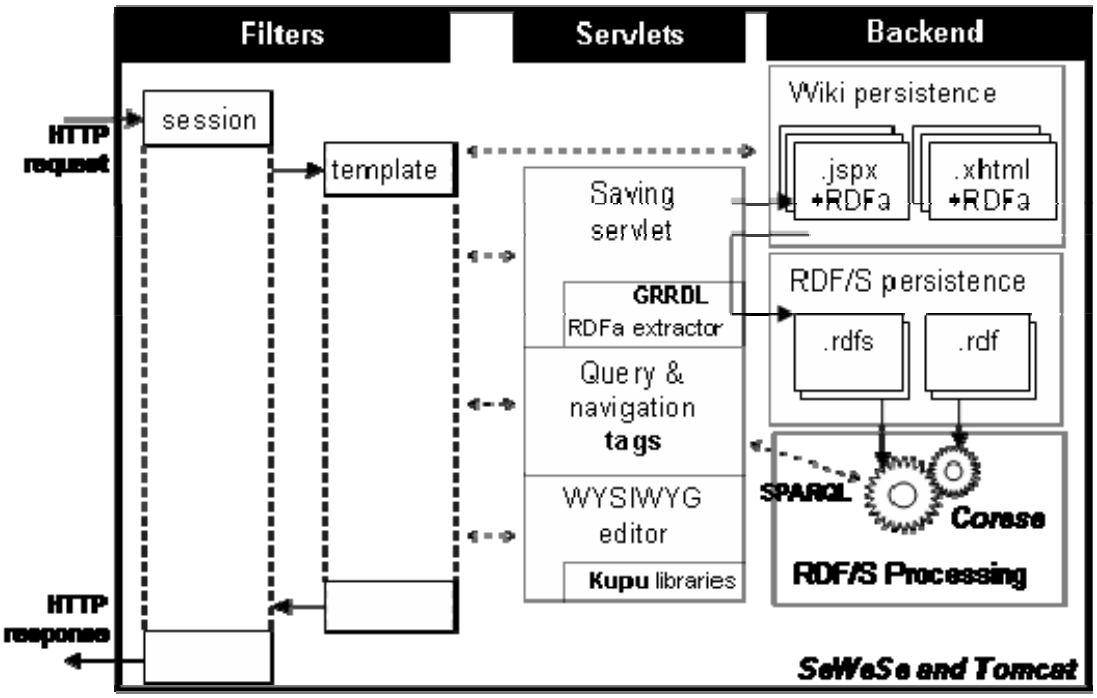

Fig 2. Architecture of SweetWiki in SeWeSe.

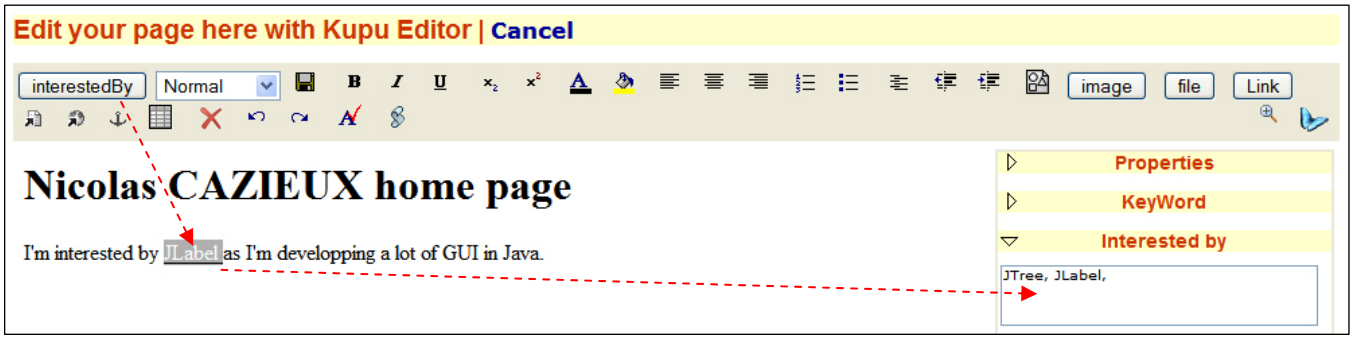

Fig 3. Editing a homepage and tagging it with personal interests.

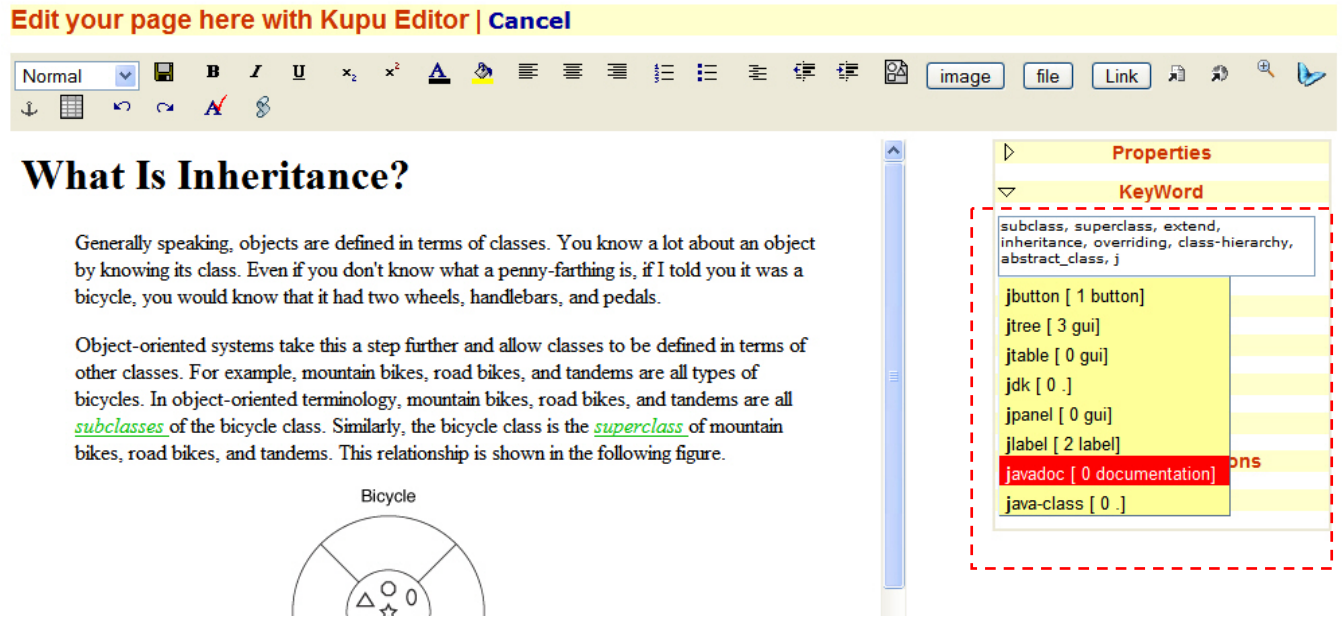

Fig 4. Tags are suggested as the user enters keywords, the number of pages using each tag is displayed and the related category.

2.3 Focus on tagging: using semantic web technology to implement a folksonomy

Like [13 and 14], we propose a mixed approach in order to "organize the tags": we link the tags together within a folksonomy described using the semantic web languages, where tags are organized in a hierarchy and related one to another using relationships like `subClassOf`, `seeAlso`, etc. Grubert goes further and proposed in [15] to define "an Internet ecology" for folksonomies *i.e.* an ontology for describing folksonomies. Like him, we believe that social tagging minimizes cost and maximizes user participation. So we do support social tagging in SweetWiki, but we also think that these tags must be organized. The system we have implemented helps users build a better folksonomy while relying on standard semantic web technologies for organizing and maintaining the folksonomy. SweetWiki uses folksonomies and social tagging as a better way to categorize the wiki documents [9, 10].

SweetWiki integrates a standard WYSIWYG editor (Kupu) that we extended to directly support semantic annotations following the "social tagging" approach. As shown in Figure 4, when editing a page, the user can *freely* enter some keywords in an AJAX-powered textfield. As the user types, an auto-completion mechanism proposes existing keywords by issuing SPARQL queries to the semantic web server in order to identify existing concepts with compatible labels and shows the number of other pages sharing these concepts as an incentive to use them.

Furthermore, related categories are also displayed in order to address the ambiguity of homonymy. With this approach, tagging remains easy (keyword-like) and becomes both motivating and unambiguous. Unknown keywords are collected and attached to new concepts to enrich the folksonomy. Later on, community experts may reposition them in the ontology, edit them, etc. The feedback coming from the tags is useful for improving the ontology.

When the page is saved in XHTML the associated metadata are saved inside using the RDFa syntax, as illustrated by Figure 5. Besides the topic tags (keywords and see also), metadata include contextual information (e.g. author, last modification, etc.). Thus the page stands ready to be served by a web server.

```
<head xmlns:sw="http://sweetwiki.inria.fr/"
xmlns:jv=" http://www.inria.fr/acacia/java-onto#">
 <meta content="JavaGui" name="sw:name"/>
 <link href="#admin" rel="sw:author"/>
 <meta content="2006-3-2" name="sw:modification"/>
 <link href="#Courses" rel="sw:hasForWeb"/>
 <link href="#JavaJPanel" rel="sw:forwardLink"/>
 <link href="#JavaJTable" rel="sw:forwardLink"/>
 <link href="[jv:GUI]" rel="sw:hasForKeyWord"/>
 <link href="[jv:JLabel]" rel="sw:hasForKeyWord"/>
 <link href="#JavaJPanel" rel="sw:seeAlso"/>
</head>
```

Fig 5. How the metadata are described in the wiki page file.

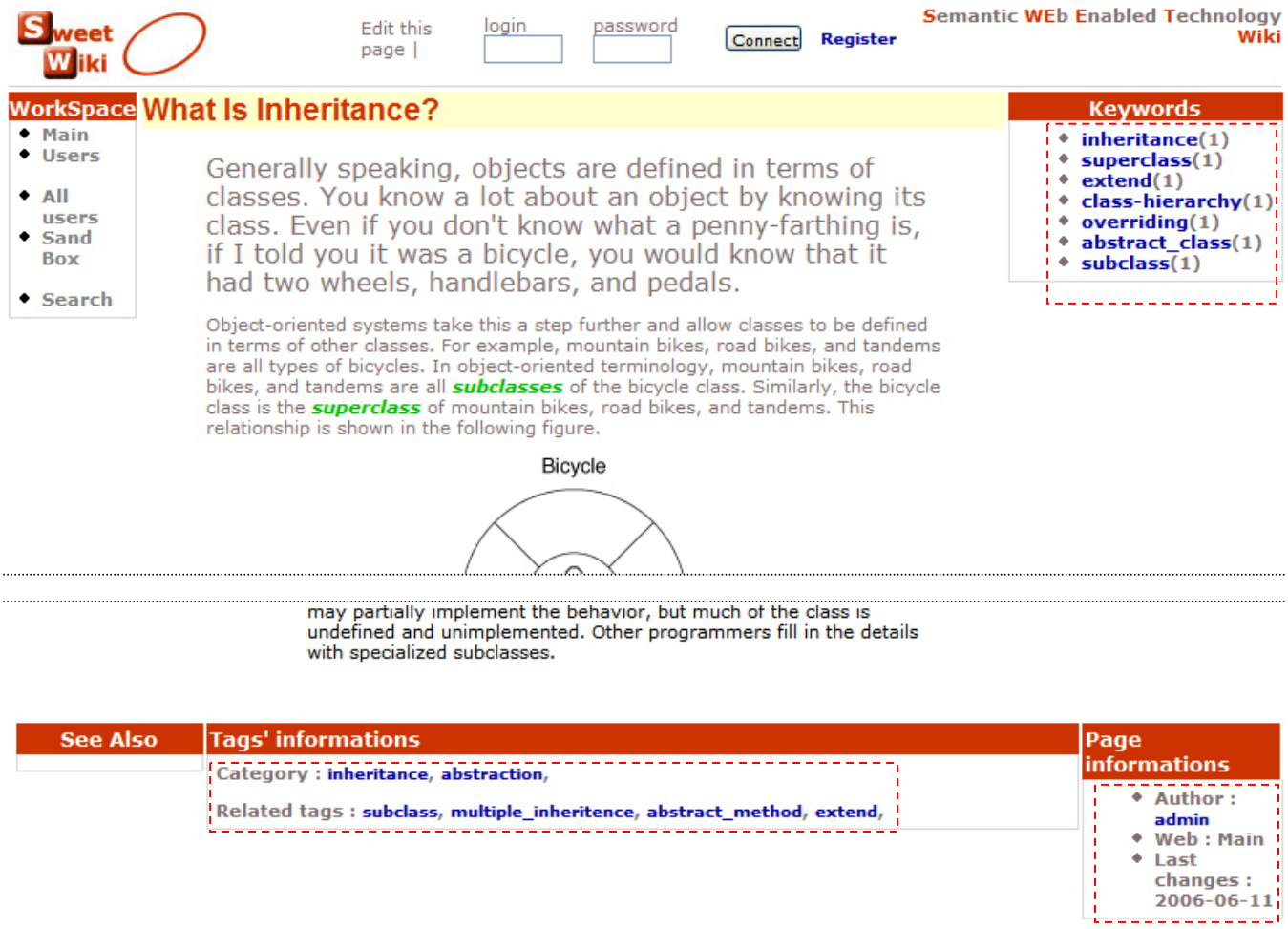

Fig 6. Faceted navigation links extracted from the tags

During the save process, the newly saved page metadata are extracted using the semantic web server API. This API uses a GRDDL XSLT stylesheet to extract the metadata in RDF/XML format and feed them to the CORESE engine. Other wiki pages that hold "create links" (links created before the current page existed) are also updated and their metadata extracted using the same process. The CORESE engine is then used to generate faceted navigation widgets: the semantics of the tags is used to derive related topics, query the engine on similar pages using SPARQL queries, etc. (see Figure 6).

The page content is saved in pure XHTML and is ready to be served (without any further translation as required with a wikiML variant). When a SweetWiki document is requested by a web browser, templates are used in order to integrate the faceted navigation widgets around the page content. These templates may be changed like the skins of TWiki for example, they are just used for decorating the final document.

2.4 Ontology editor for maintaining and re-engineering the folksonomy

Supervising tools are integrated in SweetWiki by relying on the semantic web server SeWeSe. They are used to monitor the wiki activity itself running SPARQL queries over the metadata e.g. usage frequency for tags (See Figure 7), new tags, orphan pages, etc.

In order to maintain and re-engineer the folksonomy, SweetWiki also reuses web-based editors available in SeWeSe. In our examples we tagged some Java courses, using a Java ontology. Selecting this ontology in the editor, one can add/remove/edit tags/concepts (Figure 8 and 9). In particular, if a tag/concept has been recently added it may be inserted in the hierarchy. Figure 9 shows the concept editing tool.

Using these editors, the folksonomy and the annotations may be updated. For instance, community experts can pick a couple of tags and declare semantic relations between them such as `subClassOf`. They may also merge concepts when two tags are synonymous, etc. Enhancements of the ontology seamlessly improve content sharing: search and faceted navigation benefit directly from the updates. The way the system is designed, versioning cannot break the annotations. If a tag is suddenly missing it is just treated as a new tag and if many pages exist with the old tag (pages are not touched in tag editing process), the tag would re-appear (with a high number of tagged pages, encouraging other people to use it). Re-engineering the ontology is a way of refactoring the wiki: new links appear as the ontology is enriched.

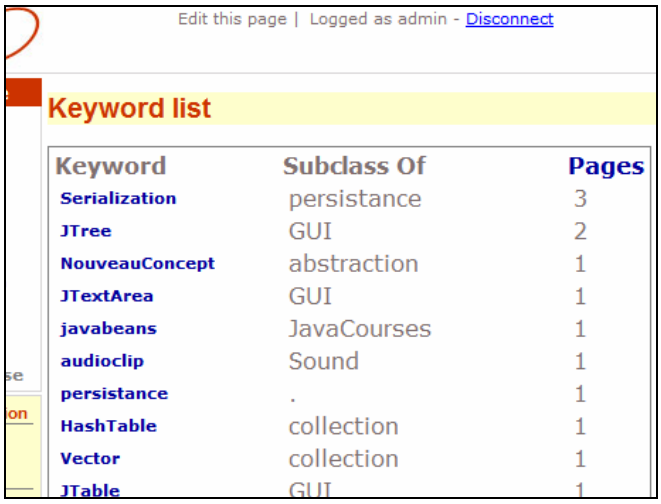

Fig 7. tags sorted by popularity

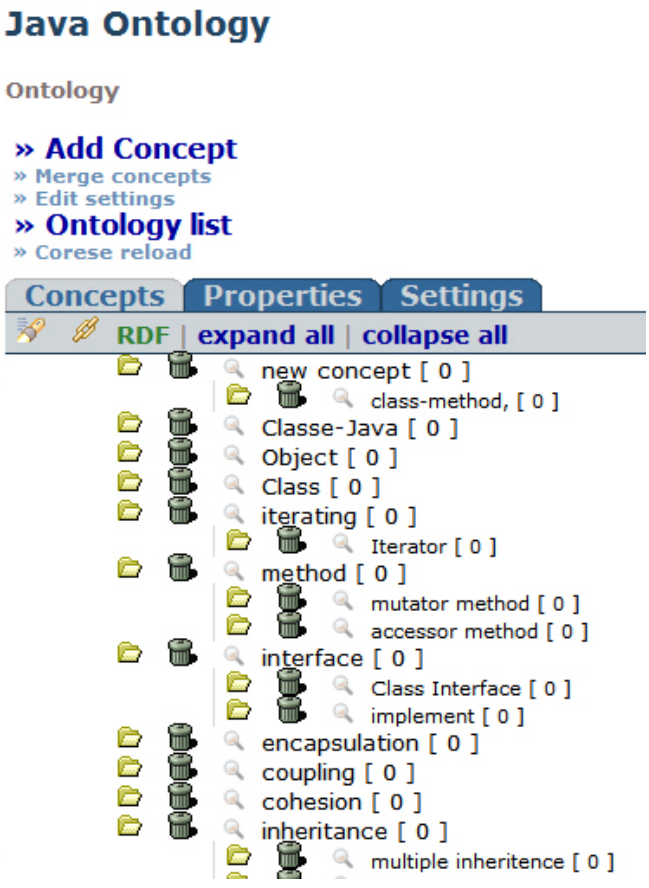

Fig 8. The ontology editor, here illustrated with the Java topics. It is possible to add/edit/merge/remove concepts and properties and even import ontologies.

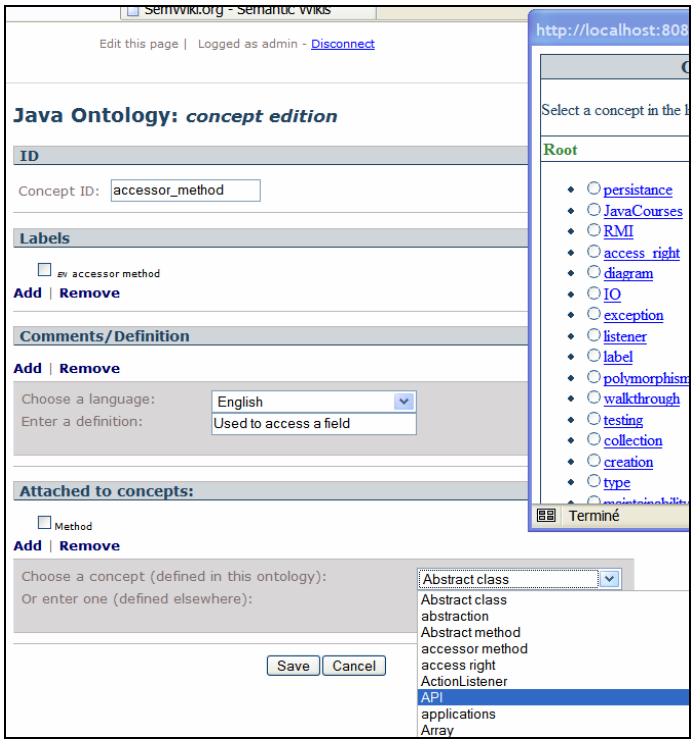

Fig 9. Editing a concept.

3. RELATED WORK AND POSITIONING

Many semantic wiki projects are being developed. Looking at the state of the art we can distinguish between approaches considering "the use of wikis for ontologies" and approaches considering "the use of ontologies for wikis" (while a few engines merge both approaches).

Most of the current projects on semantic wikis fall in the first category *i.e.* they consider wiki pages as concepts and typed links (in the page content) as relations or attributes. In this model, called a “Wikitology” in [25], the Wiki becomes the front-end of the ontology.

One of the first wikis to fall into this category is Platypus [21] that imposes separately editing the metadata for each wiki page in a “Wiki Metadata page”. It supports basic ontology editing but with no consistency check between the annotations and the ontology. It does not come with a reasoning engine and supports only basic queries. Semantic metadata are used for improving navigation but the main drawback is that the users have to switch between editing normal text and editing semantic annotations as these activities are done using two distinct text-based editors. Other wikis like SHAWN [27] offer similar features. The other wikis presented in this category address Platypus’ shortcomings by allowing semantic annotations directly in the text of the page, usually as typed links.

Rise [25] also falls in the first category: the ontology used by the community is edited via the Wiki itself and a set of naming conventions is used to automatically determine the actual ontology from the Wiki content. A proprietary language is used for describing the metadata while RDF exportation is possible. Semantic information is used for navigation and consistency

checks. The ontology is built as wiki pages are updated (rebuilt each night).

Rhizome [20] supports a modified version of WikiML (ZML) that uses special formatting conventions to indicate semantic intent directly in the page content. Pages are saved in RDF and another editor can be used to edit the RDF directly. Rhizome authors admit that this feature is dangerous as one can break the wiki behavior by entering bad RDF. To mitigate the inherent dangers of this level of openness, Rhizome Wiki provides fine-grain authorization and validation alongside the use of contexts. It is not clear how metadata improve the wiki behavior; there is no advanced search and no help for navigating the wiki so far. RDF-Wiki [29] is similar to Rhizome in that it allows RDF annotations for external processing.

SeMediaWiki [26] is based on MediaWiki. In contrast to Rise, typed links can also be used for specifying attributes of the page. For example, the following text: *San Diego is a [[is a::city]] located in the southwestern corner of [[is located in::California]]* establishes the facts "San Diego is a city" and "San Diego is located in California". While the text *Its coordinates are [[coordinates:=32°42'54"N, 117°09'45"W]]* defines an attribute named "coordinates". These data are used for faceted navigation. SeMediaWiki translates these metadata into RDF but does not use a reasoning engine yet. Other semantic extensions of MediaWiki are available such as [32] but are still at early stage of development.

Makna [31] is based on JSPWiki and provides semantic extensions as typed links. It comes with the JENA reasoning engine that allows complex queries. Its text-based editor proposes extra HTML forms (ajax-powered) for quering the semantic engine and look for concepts/properties/relationships. This is useful in the case of a large ontology.

WikSar [22, 23] enables users to enter semantic annotations from the wiki text editor using WikiWords. For example: if in a page named "PrinceHamlet", there is a line "FigureBy: WilliamShakespeare", it can be seen as a RDF statement. By combining all such embedded statements, a formal ontology emerges within the Wiki. The editor is text-based and proposes neither help of any kind to the user nor any consistency check. As pages are saved, the metadata are used to propose faceted navigation. WikSar supports queries in RDQL and SPARQL and queries can be embedded in wiki pages or templates. A distinctive feature of WikSar is the "interactive graph visualisation and navigation" tool that can be used for exploring the wiki through its metadata.

Typed links are powerful but one has to remember each concept, relation, property before typing it and this is not practical. Ace Wiki goes further: with AceWiki [40] one can add and modify sentences written using the ACE language (Attempto Controlled English [41]), through the use of an interactive Ajax-based editor. The editor is aware of the background ontology, and provides guidance to the user by proposing only valid completions. Moreover, the editor can be used to extend the ontology by creating new concepts, roles and individuals. Therefore, it is also, de facto, a simple ontology editor.

The second family of approaches focuses on "the use of ontologies for wikis". IkeWiki [24] supports both WikiML and WYSIWYG editing of page content and metadata, as well as page tagging. The editor comes with some AJAX features like auto-completion on metadata. It requires an existing ontology to be loaded. Some support for ontology editing is provided. It uses Jena and metadata are used for navigation and page rendering. Annotations can be visualized in a frame next to the wiki page. Each node is a link. IkeWiki has a nice user interface.

SweetWiki also falls into this second category. It does not implement the Wikitology model yet but we have made provision for such an evolution. So far we support the concepts of social tagging and folksonomy. SweetWiki is close to WikSar since they share many features like usage-driven ontology building, queries embedded in the wiki pages (as JSP tags), edition of metadata and page content in the same editor. SweetWiki adds a reasoning engine and an extensible WYSIWYG editor for both content and metadata, (like IkeWiki or Makna). The SweetWiki editor is AJAX-enhanced and annotating pages leads to instant gratification for users in two ways since as they type: (a) they can see an instant display of faceted links the annotation will add to the page; (b) an auto-completion mechanism proposes existing concepts from the ontology, related categories and number of pages sharing that annotation as an incentive to reuse existing tags. Furthermore, SweetWiki comes with complete user-friendly ontology supervising and editing tools. However, SweetWiki is not dedicated to *collaborative* ontology management (e.g. OntoWiki [30]) but we are currently brainstorming on how we could add such capabilities to our engine.

4. DISCUSSION

To summarize the overall scenario explored in SweetWiki, we have proposed an innovative approach that allows users to edit wiki pages and tag them using a shared conceptualization behind the scenes. In addition community experts can check the underlying model being built, look at the tags/concepts proposed by the users and (re)organize them. If this happens, annotations that users entered are not changed, but faceted navigation and search based on semantic queries are improved by new links.

As the reader may have noticed in the snapshots, our current experimentation uses an online course on Java as a test case. The learning objects are organized as wiki pages and annotated with concepts of the Java language.

A number of evolutions are currently under consideration:

- *Including forms in wiki pages*: the easy creation of pages makes it tempting to extend the concept to create small web applications in particular processing small forms. SeWeSe proposes a language merging SPARQL and JSP to generate forms from the underlying ontology. We are planning on integrating this facility to ease the development of small front-ends e.g. dedicated advanced search.
- *Natural language processing for automatic tagging*: several wikis are starting to analyze the text of wiki pages to suggest potential keywords. Seamless deduction of metadata could be achieved by applying natural language processing techniques to (semi-)automatically derive keywords from the existing content and its context.
- *Complete versioning*: support versioning of textual content, semantic annotations and underlying ontologies at the same time;
- *Collaborative management of the folksonomy*: provide groupware to assist the distributed lifecycle of ontologies;

here the wikitology approach seems only natural and we need more powerful tools to implement it efficiently.

This last point brings us back to the two-way vision *wikis for ontologies and ontologies for wikis*. This division of the current approaches is only symptomatic of the early times of semantic wikis. In the long term future semantic wikis should merge these two approaches as two facets of the same coin as some projects already started to do it; the objective being to turn this two-way vision into a virtuous circle where users maintain the ontology and the wiki *at the same time* without any artificial distinction between them. For us SweetWiki is an ideal experimentation platform to test this vision. We are just starting to experiment with the possibilities on different focus groups.

5. ACKNOWLEDGMENTS

Thanks to the student who worked on this project: Nicolas Cazieux, Gaël Crova, Adrien De Georges, Guillaume Ereteo, Claire Lecompte and Jeremy Passeron. Authors would also like to thanks the ILOG Company and the COLOR commission of INRIA for their scientific and financial support to the "Usable Intranet" project, in the context of which the work presented here started.

7. REFERENCES

[1] Stenmark, D. "Knowledge sharing on a corporate intranet: Effects of re-instating web authoring capabilities". Proceedings of ECIS 2005, Regensburg, Germany, May 2005, 26-28.

[2] Chat C. and Nahaboo, C. "Let's Build an Intranet at ILOG Like the Internet!", Proceedings of the Intraweb workshop, WWW Conference 2006, Edinburgh.

[3] Buffa, M. "Intranet Wikis". Intraweb workshop, WWW Conference 2006, Edinburgh.

[4] Buffa M., Sander P., Grattarola J.-C. "Distant cooperative software development for research and education: three years of experience", In proceedings of CALIE'04, 2004, Grenoble, France.

[5] Finck, N., Hodder, M. and Stone, B. "Enhancing Internal Communications with Blogs, Wikis, and More": http://www.nickfinck.com/presentations/bbs2005/01.html, 2005.

[6] Merrill, D. "A view into Google's inner workings", audio report about presentation at Vortex 2005: http://www.citizenvalley.org/blocnotes/index.php?id_article=241401

[7] Inside Google, Rough Type Blog, http://www.roughtype.com/archives/2005/10/inside_google.php

[8] Cunningham, W and Leuf, B. "The Wiki Way: Quick collaboration on the web". (2001). Addison-Wesley, Boston.

[9] Powers, S. "Cheap Eats at the Semantic Web Caf"é, http://weblog.burningbird.net/archives/2005/01/27/cheap-eats-at-the-semantic-web-cafe/. (2005).

[10] Hammond, T., Hannay T., Lund,B. and Scott, J. "Social Bookmarking Tools, a General Review", D-Lib Magazine, April 2005, Volume 11 Number 4, http://www.dlib.org/dlib/april05/hammond/04hammond.html

[11] Shirky. C. "Ontology is overrated". Etech Talk. http://www.itconversations.com/shows/detail470.html, (2005).

[12] Smith. G. "IA Summit Folksonomies Panel". http://atomiq.org/archives/2005/03/ia_summit_folksonomies_panel.html, . (2005).

[13] Bird. F. "Some ideas to improve tags use in social software, flat hierarchy versus categories in social software". http://fredbird.org/lire/log/2005-05-17-tags-structuration-proposal, (2005).

[14] Olsen. H. "Navigation blindness, how to deal with the fact that people tend to ignore navigation tools". The Interaction Designer's Coffee Break, Issue 13, Q1 2005. http://www.guuui.com/issues/01_05.php

[15] Gruber. T. Folksonomy of Ontology: A Mash-up of Apples and Oranges. First on-Line conference on Metadata and Semantics Research (MTSR'05). http://mtsr.sigsemis.org/, (2005).

[16] Désilet A., Paquet, S and Vinson N.G. « Are Wikis Usable? », proceedings of the 2005 International Symposium on Wikis, Oct 16-18, San Diego, California, USA.

[17] Shah, S. « The internet is Jain : How Gunslingin' Technolibertarianism Leads to Lotus Petals », in proceedings of New Forms Festival, Technography, Vancouver, BC, 2004.

[18] Cunningham. W. Ross Mayffeld's notes on Cunningham's Keynote at Wikisym 2005. "Ward Cunningham on the Crucible of Creativity". http://ross.typepad.com/blog/2005/10/ward_cunningham.html

[19] Wales. J, founder of Wikipedia / Presentation at Wiki Symposium 2005. http://recentchanges.info/?p=5

[20] Souzis. A. Building a Semantic Wiki. EEE Intelligent Systems, vol. 20, no. 5, pp. 87-91, September/October, 2005.

[21] Campanini S.E., Castagna P. and Tazzoli R. Platypus Wiki: a Semantic Wiki Wiki Web. Semantic Web Applications and Perspectives, Proceedings of 1st Italian Semantic Web Workshop. December 2004. http://semanticweb.deit.univpm.it/swap2004/cameraready/castagna.pdf

[22] Aumueller D. and Auer S. Towards a Semantic Wiki Experience – Desktop Integration and Interactivity in WikSAR. Proc. of 1st Workshop on The Semantic Desktop - Next Generation Personal Information Management and Collaboration Infrastructure, Galway, Ireland, Nov. 6th, 2005. http://www.semanticdesktop.org/SemanticDesktopWS2005/final/22_aumueller_semanticwikiexperience_final.pdf

[23] Aumueller D. SHAWN: Structure Helps a Wiki Navigate. Proceedings of the BTW-Workshop, March 2005. W. Mueller and R. Schenkel editor. http://dbs.uni-leipzig.de/~david/2005/aumueller05shawn.pdf

[24] Schaffert S., Gruber A., and Westenthaler R.: A Semantic Wiki for Collaborative Knowledge Formation . In: Semantics 2005, Vienna, Austria. November 2005.

[25] Decker B., Ras E., Rech J., Klein B. and Hoecht C. Self-organized Reuse of Software Engineering Knowledge Supported by Semantic Wikis. Proceedings of the Workshop on Semantic Web Enabled Software Engineering (SWESE), held at the 4th International Semantic Web Conference (ISWC 2005) November 6th - 10th, 2005,Galway, Ireland

[26] Krötsch M. and Vrandečić D. and Völke M. Wikipedia and the Semantic Web - The Missing Links. Proceedings of the WikiMania 2005. http://www.aifb.uni-karlsruhe.de/WBS/mak/pub/wikimania.pdf

[27] Aumueller, D. SHAWN: Structure Helps a Wiki Navigate. BTW-Workshop "WebDB Meets IR", Karlsruhe, 2005-05. http://the.navigable.info/2005/aumueller05shawn.pdf,

[28] WikiOnt: http://boole.cs.iastate.edu:9090/wikiont/,

[29] RDF Wiki: http://infomesh.net/2001/05/sw/#rdfwiki,

[30] Hepp M., Bachlechner D. and Siorpaes K. OntoWiki: Community-driven Ontology Engineering and Ontology Usage based on Wikis. Proceedings of the 2005 International Symposium on Wikis (WikiSym 2005). http://www.heppnetz.de/files/ontowikiDemo-short-camera-ready.pdf

[31] Dello K., Tolksdorf R. and Paslaru E. Makna. Free University of Berlin. http://www.apps.ag-nbi.de/makna/wiki/About, 2005.

[32] Muljadi H. and Takeda H. Semantic Wiki as an Integrated Content and Metadata Management System. Proceedings of ISWC 2005, Galway, Ireland.

[33] Merholz P. Metadata for the Masses. Adaptive Path Blog. http://www.adaptivepath.com/publications/essays/archives/000361.php, 2004.

[34] Kupu : http://kupu.oscom.org/

[35] W3C, Semantic Web Activity, http://www.w3.org/2001/sw/ et http://www.w3.org/2001/sw/Activity

[36] RDFa Primer 1.0 Embedding RDF in XHTML http://www.w3.org/2001/sw/BestPractices/HTML/2006-01-24-rdfa-primer

[37] Gleaning Resource Descriptions from Dialects of Languages (GRDDL) http://www.w3.org/2004/01/rdxh/spec

[38] Corby O., Dieng-Kuntz R, Faron-Zucker, C., Querying the Semantic Web with the CORESE search engine. In Proc. of the 16th European Conference on Artificial Intelligence (ECAI'2004), Valencia, 2004, IOS Press, p. 705-709

[39] SPARQL Query Language for RDF http://www.w3.org/TR/rdf-sparql-query/

[40] AceWiki : http://gopubmed.biotec.tu-dresden.de/AceWiki/

[41] Attempt to Controlled English (ACE) : http://www.ifi.unizh.ch/attempto/