Exploiting user gratification for collaborative semantic annotation

Davide Eynard
Politecnico di Milano
Dipartimento di Elettronica e Informazione
eynard@elet.polimi.it

Marco Colombetti
Politecnico di Milano
Dipartimento di Elettronica e Informazione
colombet@elet.polimi.it

ABSTRACT
Semantic annotations could improve the Legacy Web by adding semantics to information which has already been published in form of unstructured text. Semi-automatic annotation tools seem the most viable way to obtain a contribution from users without requiring them to have a deep knowledge about semantics, however the effort to make them work is still, most of the times, not worth the reward for using them. This paper presents a collaborative semi-automatic annotation approach for Web pages which requires almost no knowledge about semantics on the user side, but nevertheless provides an immediate advantage for the whole community: annotated data become automatically linked to a whole set of online services and resources specific to their related concepts, thus providing an instant reward for users in the form of additional available information.

INTRODUCTION
One of the biggest challenges for the Semantic Web community is trying to add semantics to information which has already been published in the form of unstructured text. Many approaches have been tried to add semantics to unstructured pages, and the idea of annotating Web contents seems a good one, allowing for a real "read-write Web" where any user or machine can add metadata to any piece of information.

While both automatic and manual annotation systems still present some open issues (just to name few, word disambiguation[1] for the former ones and lack of precision[2] for the latter ones), the semi-automatic approach currently seems the most feasible. However, at the present time even this kind of systems still lack that wide acceptance that would make a semantic annotation system really useful on the World Wide Web.

Our project starts with the assumption that one of the possible reasons of this is that these systems do not provide enough advantages to motivate the user efforts to make them work. Trying to overcome this problem, we decided to increase user motivation building an easy and rewarding annotation tool.

To do this, we envisioned a collaborative semi-automatic annotation approach for Web pages, which allows to connect pieces of unstructured text with standard concepts without requiring specific knowledge about semantics; as a result, annotated data become automatically linked to a whole set of services and resources specific to their related concepts, thus providing an instant reward for users in the form of additional available information.

After a description of the related work, we show our approach and describe the architecture and the implementation of our prototype. We then describe current and planned evaluations for the tool, showing its main pros and cons. Finally, we conclude with a summary and a discussion on future work.

RELATED WORK
Several tools and approaches exist to create annotations of both Web resources and abstract concepts: [3] provides a survey of the main semi-automatic annotation platforms, while [4] presents a unified formal model, able to describe and integrate annotations inside traditional documents, semantic wikis, semantic blogs and collaborative tagging systems.

Our project shares one basic principle with collaborative tagging: users annotate for themselves, but the system automatically shares personal annotations between users so that everyone contributes to the overall value of the system. At the same time, the differences between collaborative tagging systems and ours are rather strong: first of all, the granularity of our semantic annotations is much higher, as it involves single words inside a Web page instead of more complex kind of resources; moreover, users in our system cannot annotate using unconstrained strings, but they have to choose one concept from a list of suggested ones. This last point is particularly important, as it defines a completely different annotation paradigm: while tagging is bottom-up and flat, our semantic annotation is top-down and hierarchical, with all its advantages and limitations[5].

Annotea[6] is a semantic annotation tool which enhances collaboration via shared metadata based Web annotations, bookmarks, and their combinations. It uses an RDF based annotation schema for describing annotations as metadata and XPointer[1] for locating the annotations in the annotated document. Different client softwares for Annotea have been built: between these Annozilla[2], created as an extension of the Mozilla browser.

KIM[7] is a software platform for automatic annotation, indexing and retrieval of information. Its approach is based

[1] `http://www.w3.org/XML/Linking`
[2] `http://annozilla.mozdev.org/`

on the assumption that *named entities*[8] have to be handled in a special way, as they denote particulars (individuals or instances) while other words denote universals (concepts, classes, relations, and attributes). It then uses NLP to recognize and identify them inside a text, with respect to a predefined ontology.

MnM[9] provides both automated and semi-automated ontology driven support for annotating Web pages with semantic contents. The annotations are written as markup inside a document. Magpie[10] uses an ontology infrastructure to semantically markup Web documents on-the-fly. Both of these tools work as browser extensions and provide new pieces of information related to the annotated text, but both seem to get these information just from the ontology, and not from external data sources.

Revyu[11] is not a generic annotation tool, but rather a reviewing and rating Web site. We consider it related to our work for its attention towards Linked Data principles and best practices [12, 13]. This system does not only work as a service usable by humans, but also provides information in a reusable format that can be easily integrated with other data.

Gnosis[3] is probably the project which is most similar to ours. It works as a Firefox extension and when a Web page is loaded inside the browser it immediately locates key information such as people, organizations, companies, products and geographies hidden within the text. It then highlights these concepts in the page and provides links to specific search engines which change depending on the resource type. One of the main drawbacks of this tool is that it relies on NLP and on a read-only knowledge base, both for what concerns named entities and for the search engine list. While this is surely convenient for kickstarting the application, active user participation could make the tool more precise and powerful; moreover, it could help to disambiguate strings that match many different concepts.

PROJECT OVERVIEW

We started our project with the assumption that one of the possible reasons why many semantic annotation systems still do not have a wide acceptance between Internet users is that they do not offer them much, or at least not much enough to motivate their efforts for using them. The question we try to answer is: how can we exploit the concept of gratification to make users semantically annotate unstructured text?

To the best of our knowledge, available semantic annotation tools currently provide some kind of reward to the user in the form of additional, concept-specific information that is accessible when a piece of text is annotated as being an instance of one particular concept. However, this information is usually taken from the ontology used by the annotation system, and as a result it is very schematic and constrained by the ontology itself.

As an example, think about a user who is browsing a Web page containing a review about one of the Harry Potter books. The user might be interested in knowing something more about these books, so why should she annotate "Harry Potter" as being a book? No ontology is currently able to provide the same quantity of information that can be found on the Web about this topic, and probably if one was built to do so it would not be able to keep the pace of the ever growing Web. However, it is still possible to link this concept, once we know it's a book, to a huge number of services, data sources and search engines which will provide huge quantities of related information: as an example, Figure 1 shows results from some book-related services, ranging from textual descriptions to RDF data, from user ratings to the complete books in PDF format.

The purpose of our project is to build an annotation system able to provide this kind of experience to the user: the additional context-related information should be accessible through a template page like the GeoHack page [4] in Wikipedia, and could provide links to specific services, search engines and query strings[5]. As a result, we'll be able to use user communities and Semantic Web technologies in a virtuous cycle: on one side we'll exploit user spontaneous collaboration to increase the amount of semantic metadata available; on the other side, we'll use these new data to make the system more rewarding, thus encouraging user participation.

Users

The main goal is to make users spontaneously contribute to the system with semantic annotations. This translates in two main requirements: motivate users and make their contributions useful by allowing them to be visible and reusable in the "Web of data".

One way to increase motivation in users is by giving them some kind of reward for their participation[14, 15]. This is the reason why our system is designed to provide it in a double way:

- an instant gratification, by automatically linking the annotated data with additional sources of information,
- and a long term one, by sharing annotations in a standard and structured way so they can be searched, accessed and linked with other data.

Keeping the average Internet user in mind, we also have to face the problem of identifying what kind of related information might be considered interesting enough for a particular community of practice[16]: for instance, an adult does not have the same needs as a teenager, and people from different parts of the world might want to access different local services instead of more general ones which are available worldwide. For this reason, these links should be collected in pages which are very easy to create, edit and share such as inside a wiki system: anyone has a page by default for a particular concept, but he can customize it or directly choose another one.

[3] `http://gnosis.clearforest.com`

[4] Visible, for instance, clicking on the geo coordinates at `http://it.wikipedia.org/wiki/Milano`

[5] Like the "search webbits" at `http://www.searchlores.org/rabbits.htm`

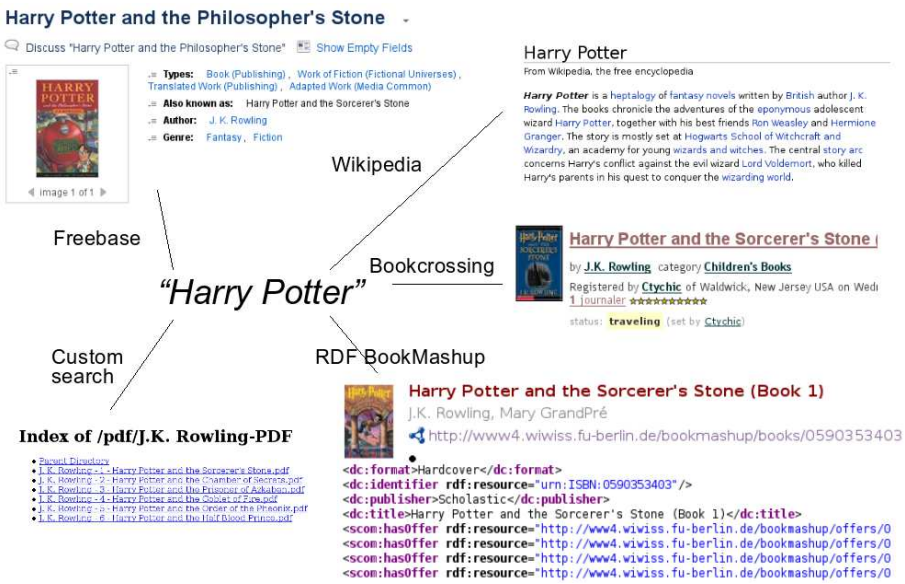

Figure 1. Some possible links for "Harry Potter", when considered as a book.

Finally, the system has to be easy to use and intuitive: users don't have to know specific details about semantics, but they just have to choose a concept from a list of suggested ones. Also, the only part of the system they have to use is a browser extension, which adds a new option to the contextual menu that appears when the right mouse button is clicked (following the affordance of this button).

Data

Making semantic metadata reusable by other applications basically means publishing them in RDF and making them available through a SPARQL endpoint. A particular attention has been devoted to using common ontologies for interoperability with other systems, and providing a mapping system to automatically translate internal terms with terms from well known RDF vocabularies.

Whenever a term is tagged as being instance of a particular concept, additional information can be automatically harvested from the Internet to make the model richer, and some automatic annotations (such as the ISBN for a book, such as in [11], or an IMDB id) could be added to the knowledge base.

PROJECT ARCHITECTURE AND IMPLEMENTATION

The architecture of our system is shown in Figure 2: its main modules are the client extension, the knowledge management tool and the annotation server.

The client

The client application is nothing more than a normal browser with a plugin. The user just has to select some text and click the right mouse button, choosing the contextual menu option "Speakin' about": a popup window then appears, allowing her to choose the name of the concept related to the selected string.

Actually, as soon as the menu option is chosen the browser extension communicates with the knowledge management tool (KMT), sending the selected string and asking for possible concept suggestions. The KMT replies with its suggestions (see next paragraph for details) and when the user chooses one it collects all the information needed for the annotation.

Once the annotation is saved, the selected text becomes a link to a special page which contains a collection of concept-related links that can be followed to find new pieces of related information.

The Knowledge Management Tool

The KMT manages the communication with the annotation server, saving the metadata when it receives them from the user. Also, it provides additional functionalities thanks to its three submodules: the user base, the concept modules and the template engine.

User Base

As everyone can write any kind of annotation, at least an authentication mechanism is needed to bind annotations with users. This way, users can either see everyone else's annotations on one page or override them with their own ones. Additional features which might be useful are a social network to connect users and a trust system to allow users to allow (or deny) by default someone else's annotations.

Concept Modules

As users submit their strings, they should be shown some concept suggestions: the concept modules take care of this, searching for the string inside ontologies, Web sites, and online services to find a matching concept to suggest.

The structure is modular as many different approaches can

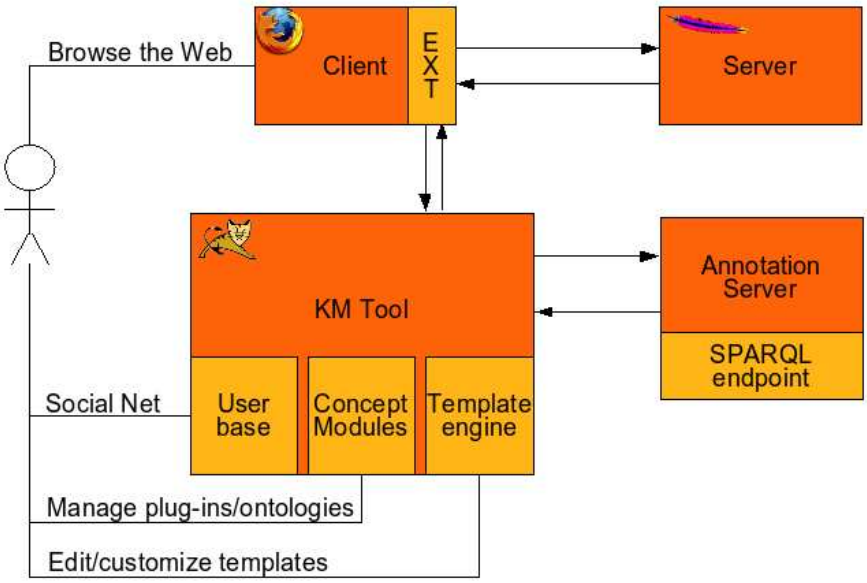

Figure 2. The system architecture.

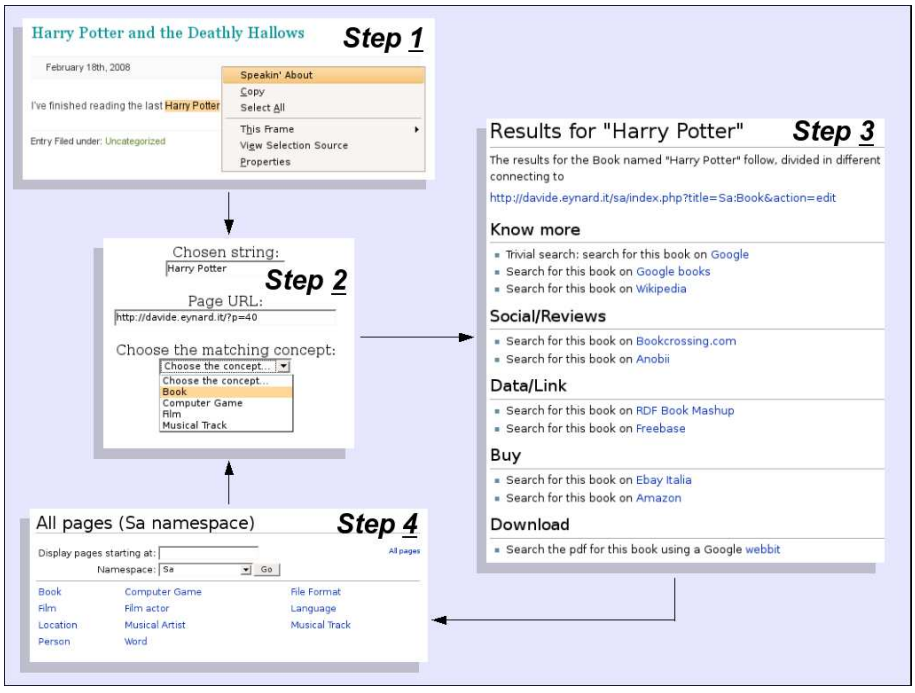

Figure 3. A usage example of the tool, from the context menu to the search page.

Annotea property	Value in Speakin' about
rdf:type	Annotation (as defined in Annotea)
annotates	The URI of the annotated resource
context	XPointer of the annotated piece of text
body	Could be left empty, or used to correct/disambiguate annotated text
dc:creator	Username of annotation creator
created	Date of creation
dc:date	Date of last update
related	URI of the related concept

Figure 4. Matching between Annotea properties and annotation values used by Speakin' about.

Class	Gnosis	Speakin' about
City	6	5
Company	2	2
Continent	1	1
Country	18	17
Industry Term	3	0
Organization	6	4
Person	15	10
Product	1	1
Province or State	3	3

Figure 5. Comparison between the number of concepts automatically found inside the main English Wikipedia page by Gnosis and the ones also recognized by Speakin' about.

be taken: for instance, a module we developed in our prototype searches for matching concepts or instances inside some domain ontologies; another one searches inside Freebase[6]; another one could use Wikipedia, or a search engine for ontologies like Swoogle. As any module might suggest concepts with different names, this component also comes with a mapping service which allows to map these names with some fixed concepts inside one main vocabulary (which, basically, is the ontology of the concepts for which a special page already exists).

Template Engine

Special pages with related links exist as templates, that is pages that receive the selected string as a parameter and use it to build all their links (as an example, check the wiki code of the GeoHack page described above). The template engine is the system which allows to manage templates and matches them with concepts. It could be very simple (a database which contains the matches, plus some HTML files) or more complex: in our case, we chose to manage the templates with a wiki-like system, allowing users to easily create, edit and share them. In this way, the possibilities offered by the application are not constrained by anyone and can grow thanks to user contributions.

The Annotation Server

The information contained in an annotation made with our tool basically connects (at least) the annotated URL, the string that represents the name of the concept instance, the matching concept and the name of the user who's saved the annotation. The annotation server is in charge of storing this information and making it available in a standard format.

Designing our system we decided to consider the annotation server as a separate module: the reason was that we wanted to build our system on already available and consolidated technologies, rather than programming everything from scratch. For instance, Figure 4 shows how we could save our metadata in an Annotea server, following its ontology specification[6].

Implementation

Currently we have a very simple prototype, developed as follows: the browser extension is a Firefox extension programmed in Javascript; the KMT has been written in Java and runs as a servlet on a Tomcat server; the annotation server is a light application written in Java, which saves its information inside a SQL database end exports it in RDF; thanks to Joseki, it can then be queried as a SPARQL endpoint; the browser extension communicates with the KMT via HTTP, and the KMT communicates with the annotation server via RMI.

[6] `http://www.freebase.com/`

SYSTEM EVALUATION

The tool has been designed with ease of use in mind: for this reason, the whole process of semantic annotation reduces to just few steps. At the first step (see Step 1 on Figure 3), the user selects some text from the current Web page, clicks the right mouse button and chooses the "Speakin' about" option. Then (Step 2) she is shown the chosen string, the originating page and a list of suggested concepts. When the user chooses one of the concepts and submits the information, the annotation is done and the Web page is updated with a new link on the annotated text. If the user clicks on that link, she is shown the search page (Step 3), containing the list of search engines related to the particular concept chosen by the user. Of course, once the user understands how wiki templating works se can change existing search pages or create new ones, allowing the concept selection tool to provide more choices (Step 4).

As the system is inherently dependent on user participation and, at the same time, it aims at linking information from different sources, it opens to two different evaluation approaches: on one side an evaluation of usability and user satisfaction, and on the other one a test on the exported data and how well it integrates with heterogeneous sources. Being still in the prototypal phase it was not possible to test the system with a realistic user base, however we were able to perform user-independent tests on it. For instance, we verified that the application was actually able to provide users with the promised additional information and links: to do this, we decided to compare our results with the ones provided by Gnosis. We performed the test on the main English Wikipedia page, checking how many concepts were automatically detected by Gnosis. Then we passed the matching strings to our Knowledge Management Tool to see if it was able to suggest the same concepts. For this task we used the Freebase module, which relies on the vast amount

of knowledge harvested from Wikipedia and enriched by its user community.

The results are summarized inside Figure 5: Speakin' about is able to recognize almost all of the concepts detected by Gnosis, however it's much more sensible to typos (which is the reason why it could not detect one City and some names); moreover, it's not able to recognize industry terms like "bank" or "environmental law" as they're not named entities and they do not appear in Freebase. Conversely, Speakin' about offers a larger taxonomy, which is the one provided by Freebase: in the same page, it was able to detect football teams, book titles, actors, and so on. Also, similar concepts are suggested based on partial matches of the selected strings; more concepts are provided at the same time, so that there are more chances for users to disambiguate the term; finally, the related search engines provided by Speakin' about are usually much more than the ones provided by Gnosis, and their number can grow thanks to user contributions.

For what concerns data evaluation, we have put our effort in following Linked Data recommendations, with the purpose of making our metadata public and easily available to other applications. Our prototype system is already able to export its information in RDF and provides a SPARQL endpoint to allow queries over the knowledge base. It is thus possible to ask for all the pages which "speak about" some concept (or some instance), and link this information with other data to answer more complex queries: for instance, importing an ontology about cinema, a user can ask for all the pages that speak about movies in which a particular actor has starred.

Other collateral advantages spawn from annotating information with Speakin' about: first of all, the system allows not only to add semantic metadata about URIs (that is, saying that a page is about some particular concept), but also to map strings with specific concepts, disambiguating them and offering access to a wealth of concept-specific services (see an example of disambiguation in Figure 6, where "Verdi", which in Italy is a color, a composer and a political party, is identified as a composer). In particular, users instantly have access to new pieces of linked information, which grow thanks to collaboratively edited templates, harnessing the power of available services and specialized search engines. Some template pages (ie. books and movies), containing links to specific services and search strings, are already available as a proof of concept, and thanks to a wiki system users can easily edit them and create new ones.

CONCLUSIONS AND FUTURE WORK

In this paper we described a collaborative semi-automatic annotation approach for Web pages, which allows users to connect pieces of unstructured text with related concepts. As an immediate result of these annotations, users are provided with related information about annotated concepts, in the form of pages containing links to concept-specific services and search string. These pages are built up from templates which can be created, modified and shared by the community inside a wiki-like system.

Figure 6. Our prototype in action: thanks to the ontology plugin, the ambiguous string *Verdi* is disambiguated thanks to its association with the *composer* concept. In this case, as the composer name is already present inside the ontology, the concept instance *Giuseppe Verdi* is suggested to the user.

The novelty in our approach is represented by the following aspects: first of all, the additional information about annotated concepts is not provided by the system itself but it's harvested from the Internet, taking advantage of all the systems which freely share their data; then, all the saved metadata are made public and easily available to other applications, as they are saved in RDF and exposed through a SPARQL endpoint; finally, the whole system relies on user participation and uses the new linked information as a reward for the users.

Our application is currently in a prototypal form. As a future work, we plan to build a system complete in all its modules (in particular, the user management and the connection with an Annotea server) and make it available on the Internet: this will allow us to complete our evaluation with a real user base, in terms of participation and user feedbacks.

REFERENCES

1. L. Reeve and H. Han. Semantic annotation for semantic social networks using community resources. *AIS SIGSEMIS Bulletin*, 2(3&4):52–56, 2005.

2. M. Erdmann, A. Maedche, H. Schnurr, and S. Staab. From manual to semi-automatic semantic annotation: About ontology-based text annotation tools. In P. Buitelaar and K. Hasida (eds.), editors, *Proc. of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content*. Morgan Kaufmann, August 2000.

3. Lawrence H. Reeve and Hyoil Han. Survey of semantic annotation platforms. In Hisham Haddad, Lorie M. Liebrock, Andrea Omicini, and Roger L. Wainwright, editors, *SAC*, pages 1634–1638. ACM, 2005.

4. Eyal Oren, Knud Möller, Simon Scerri, Siegfried Handschuh, and Michael Sintek. What are semantic annotations? Technical report, DERI Galway, 2006.

5. Clay Shirky. Ontology is overrated: Categories, links, and tags, 2005. http://www.shirky.com/writings/ontology_overrated.html.
6. Jos Kahan and Marja-Ritta Koivunen. Annotea: an open rdf infrastructure for shared web annotations. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 623–632, New York, NY, USA, 2001. ACM Press.
7. Atanas Kiryakov, Borislav Popov, Ivan Terziev, Dimitar Manov, and Damyan Ognyanoff. Semantic annotation, indexing, and retrieval. *Journal of Web Semantics: Science, Services and Agents on the World Wide We*, 2(1):49–79, 2004.
8. Nancy A. Chinchor. Proceedings of the Seventh Message Understanding Conference (MUC-7) named entity task definition. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, page 21 pages, Fairfax, VA, April 1998. version 3.5, http://www.itl.nist.gov/iaui/894.02/related_projects/muc/.
9. M. Vargas-Vera, E.Motta, J. Domingue, M. Lanzoni, A. Stutt, and F. Ciravegna. MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pages 379–391, Siguenza, Spain, 2002.
10. John Domingue, Martin Dzbor, and Enrico Motta. Collaborative semantic web browsing with magpie. In Christoph Bussler, John Davies, Dieter Fensel, and Rudi Studer, editors, *ESWS*, volume 3053 of *Lecture Notes in Computer Science*, pages 388–401. Springer, 2004.
11. Tom Heath and Enrico Motta. Revyu.com: A reviewing and rating site for the web of data. In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudr-Mauroux, editors, *ISWC/ASWC*, volume 4825 of *Lecture Notes in Computer Science*, pages 895–902. Springer, 2007.
12. Tim Berners-Lee. Linked data. World wide web design issues, July 2006.
13. Chris Bizer, Richard Cyganiak, and Tom Heath. How to publish linked data on the web, 2007.
14. Eyal Oren. Semperwiki: a semantic personal wiki. In Stefan Decker, Jack Park, Dennis Quan, and Leo Sauermann, editors, *Proceedings of the 1st Workshop on The Semantic Desktop at the ISWC 2005 Conference*, pages 107 – 122, Galway, Ireland, November 2005.
15. Dan Bricklin. The cornucopia of the commons: How to get volunteer labor, August 2000. http://www.bricklin.com/cornucopia.htm.
16. Etienne Wenger. *Communities of Practice. Learning, meaning, and identity*. Cambridge University Press, New York, Port Chester, Melbourne, Sydney, 1998.