An Ontology-based P2P System for Query-based Semantic Annotation Sharing

Norberto Fernández-García, Luis Sánchez-Fernández,
José M. Blázquez-del-Toro, David Larrabeiti

Telematic Engineering Department.
Carlos III University of Madrid.
{berto,luiss,jmb,dlarra}@it.uc3m.es

Abstract. The semantic annotation of billions of current Web resources is one of the core challenges for building the Semantic Web. In the state of the art several useful semantic annotation tools can be found. But, as far as we know, no one of these systems exploits the force of the millions of users which daily look for information on the Web. In this paper we introduce the SQAPS *Semantic Query-based Annotation, P2P Sharing* system, which aims at providing a mean of defining and sharing query-based semantic annotations. In order to share such annotations in a computer understandable manner, we propose the usage of an ontology-based P2P network.

1 Introduction

The current Web is an enormous repository with billions of linked pages. Most of these pages contain information in natural language, easy to understand by humans but not by computers, so automatic handling of their information is difficult.

In order to ease the automatic information processing, we need to describe current natural language Web resources in a *computer-understandable* manner. Making Web resources understandable by machines requires to add *semantic data* to such resources, to *semantic annotate* them.

The semantic annotation of billions of current Web resources is one of the core challenges for building the Semantic Web [1]. In the literature we can find several proposals of tools and systems for semantic annotation of Web resources, but, from our point of view, none of these systems, takes advantage of the force of the millions of users who every day look for information on the Web.

In this paper we introduce the SQAPS, *Semantic Query-based Annotation, P2P Sharing*, system. The main idea behind this system is to exploit keyword-based user queries in semantic annotation of Web resources. Instead of annotating directly Web resources, as most of current systems in the state of the art suggest, we propose a system in which users annotate their queries. The keywords in these *semantic* or *annotated* queries are sent to classical Web search engines, obtaining a vector of URLs as a result. While browsing these results, a user can say if a certain resource is relevant to his query. If so, we can associate the Web resource with the semantic query, giving implicitly a certain semantic to the Web resource. As we expect that software applications, and not only humans, access and use these semantic resource annotations, we have formally described such annotations using a lightweight RDFS [2] ontology. Concrete annotations of Web resources will be instances of a class defined in such ontology, which will be represented using RDF [3].

In order to be useful, the personal semantic resource annotations generated by the SQAPS system, need to be shared with other SQAPS users. In order to do so, and following approaches as [4], [5], [6], we propose the usage of a P2P network. Having this into account, our SQAPS system could be defined as an ontology-based P2P system for query-based semantic annotation sharing.

The rest of this paper is organized as follows: next section briefly describes some related works from the world of semantic annotation. Section 3 introduces the SQAPS system, describing briefly its architecture and working model, the SQAPS ontology, and some implementation ideas. Section 4 introduces, with discussion purposes, a set of issues related with SQAPS system. Concluding remarks in section 5, finalize this paper.

2 Related Work

The field of semantic annotation has been an active area of research for long time. In the state of the art we can find several interesting proposals, like for example: SEAN [7], a system for automatic semantic annotation of content-rich template-based HTML documents; SemTag [8] designed to provide automatic semantic annotation of large amounts of documents, using information obtained from TAP [9] knowledge base; Annotea [10], a manual, annotation system which allows users define annotations of XHTML or XML resources and share such annotations using web-based annotation servers; AeroDAML [11] which automatically annotates documents using natural language processing technologies; CREAM [12] which allows users to manually generate annotations of Web resources by typing, by selecting pieces of text from these resources, or by associating to the resource elements in a knowledge base; S-CREAM [13] evolution of CREAM to allow semiautomatic annotation of resources using natural language processing techniques; PANKOW [14] an automatic system which uses linguistically-based regular expressions, and statistics from Google [15] queries, to identify instances of a concept in a text; SMORE [16] which allows the manual semantic markup of HTML documents using ontologies as knowledge sources; the COHSE Annotator [17] which allows users to select text in a resource and associate to such text a concept or instance in an ontology; MnM [18] which allows the automatic or semiautomatic annotation of text-based Web resources using natural language processing tools; or the SHOE Knowledge Annotator [19] which allows the addition of annotations in SHOE language [20] to HTML documents. But, as far as we know, no one of these systems takes advantage of the force of the millions of the users who every day look for information on the Web.

They all propose the direct annotation of Web resources, instead of the indirect annotation, by association of an annotated query, proposed by SQAPS system. Another work which deserves special attention is [21], where the authors propose a system for deep annotation. In that system, SQL queries to a database, used to generate dynamic Web pages, can be annotated by the Web site provider. But, as far as we know, this system does not exploit the annotation of keyword-based user queries in annotating Web resources, as proposed by SQAPS system. Sharing the annotations in a P2P manner is also not suggested.

3 The SQAPS system

3.1 System Architecture and Working Model

Figure 1 shows the intended architecture of the SQAPS system. The main components of our system are:

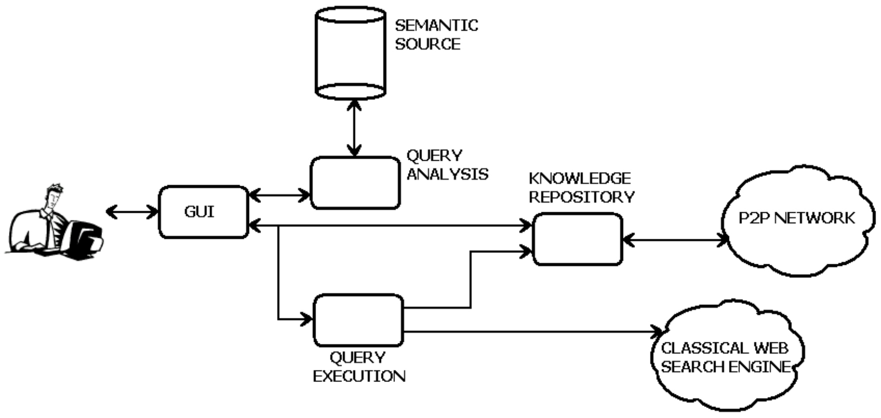

Fig. 1. SQAPS System Architecture

Query Analysis Its main purpose is to allow the annotation of a keyword-based query by the user. In order to do so, the Query Analysis component divides the query into candidate terms, each of which consisting of a word or sequence of words of the query, which can represent at least a unit of meaning. In order to associate terms and meanings, and decide what are the pairs [term,meaning] interested to the user purposes, the Query Analysis component uses the information of a Semantic Source, and asks to the user about his interests.

Semantic Source Intended to provide concrete meanings to terms. In our context, a Semantic Source should provide at least a list of terms, and for each term, a set of possible meanings with their human-readable descriptions. A unique identifier for each pair [term, meaning] should also be provided.

Knowledge Repository It stores the semantic annotations defined by the user, and also a part of the annotations that other users have defined and shared. This Knowledge Repository can be accessed by local user, but also by other peers in the SQAPS P2P network. It also can access remote repositories of other peers.

Query Execution Receives as input the annotated user query and looks for relevant resources in the Knowledge Repository. If results are found, these are shown to the user, which can decide to look for more information or not. If no results are found, or the user requires more information, this module takes the keywords from the annotated query and sends such keywords to a classical Web search engine. The results are shown to the user, who can annotate a resource by clicking on a button. By doing so, the user states that the resource is relevant for the annotated query and a pair [URL, RDF document representing the semantic query], is inserted into the Knowledge Repository and the annotation sharing P2P network.

P2P Network Main functionality of SQAPS system is semantic annotation creation and sharing. For this purpose, the basic functionality of the P2P network will be to allow annotation sharing. In our system, annotations are associations of URLs representing Web resources and RDF documents representing the semantic annotations of such resources. With these requirements, a good approach could be to rely on Distributed Hash Table (DHT) P2P networks, which offer good performance both in scalability and response time [22]. The hash of the URL of the resource being annotated would be used to decide which peer/s should store the annotation, and later, to retrieve the annotations related with a certain resource.

In order to clarify the purposes and operation model of all these components, we provide a basic example of the intended annotation process of our application. It consists of the following steps:

1. The process starts with a user writing a keyword-based query. For instance, let's assume that the user is looking for information about Caml programming language, and he types as a query *Caml programming language*.
2. This textual user query is sent to the Query Analysis block, which divides such query into candidate terms. For instance, the candidate terms originated in the analysis of our example query are *Caml*, *programming*, *language*, *Caml programming*, *Caml language*, *programming language* and *Caml programming language*. Sections of query between quotes are treated as a single terms and not divided. As some studies suggest that typical user queries are not too long (only a 25% with three or more words) [23], we expect that the number of terms will not be so big.
3. Once the system has the candidate terms, it looks into the Semantic Source what are the possible meanings associated to such terms. If candidate terms

are found in the Semantic Source, a list of possible meanings is shown to the user. For instance, in the testing prototype that we are currently developing, the Semantic Source consist of a WordNet 1.7 [24] lexicon. In such source, terms *Caml, Caml programming, Caml language* and *Caml programming language* are not included, whereas terms *programming* (2 senses), *language* (6 senses) and *programming language* (1 sense) are included. With these results, the system displays to the user the possible interpretations of the found terms. The list of interpretations can be ordered taking into account the Knowledge Repository information, which reflects user interests. The idea is to make the decision of user easier, including as first list items those which are expected to be more relevant for user's interests. For instance, if we look into WordNet 1.7, we can find two senses for the term *programming*:
   - Setting an order and time for planned events (scheduling).
   - Creating a sequence of instructions to enable the computer to do something (computer programming).

   If we assume that the user is a computer scientist and used to look for information about programming languages, it is possible that he had used the term *programming* with the second sense in past queries/annotations. In such a case, this pair [term, sense] could already be in the Knowledge Repository. Then, the senses of the term *programming* would be ordered so that the first one to be shown to user would be the one more related with the expected user's interests.
4. Given the list of possible terms and meanings, the user selects what combination better reflects his intention, annotating the query. Several possibilities may arise here:
   - No one of the terms appears in the Semantic Source, or the senses included there are not good to reflect user intentions. In such a case, the user can simply decide to not annotate the query. This unannotated query is sent to the Query Execution component, which simply forwards the query to a classical Web search engine and returns the results.
   - Part of the terms are found, and the other ones are not found, or the meanings in the Semantic Source do no reflect user intentions. Then the result will be a partially annotated query. A problem which might arise in this situation, is that it could give origin to alternative query interpretations. For instance, let's assume that the user has typed the query *Caml Light programming language.* The term *programming language* is found in the Semantic Source and the user annotates it, but he tells nothing about the other query components. Then the problem comes from the fact that the system can interpret the rest of the query in two different ways: as having two different terms, *Caml* and *Light*, or as having only one, *Caml Light.* At the moment the approach we are following is to use the information as provided by the user: if he has typed *Caml* and *Light* as different words, they are interpreted as different terms, and if he has typed *"Caml Light"* a single term will be generated. Other approaches, like for example analyzing the textual contents of the resource being annotated, in order to find which of the interpretations is the most frequent

in such contents, might be explored in the future. Another aspect which should be noted here, is that though some of the terms of the query are annotated and some others not, we will store all the terms in the final annotation. The reason for such decision is that we expect that human users can view annotations of resources, and we think that removing unannotated terms could produce a lost of context information, which could be useful for such users.
   - All the terms are found and annotated by the user.

   In the last two cases, the result will be an annotated query. Let's assume that this is our situation, and the user has generated an annotated query composed by terms *Caml* (not annotated) and *programming language*.
5. The annotated query is then sent to the Query Execution system. This system first looks into the Knowledge Repository in order to find possible useful resources for user interests, and returns the results. The Knowledge Repository component looks for information in its local contents and if results are found, these are shown to the user as a URL list. If no results are found, or the user requires more information, this module takes the keywords from the annotated query (that is, the sequence of words *Caml programming language* in our example) and sends such keywords to a classical Web search engine. The results of this engine are shown to the user as a list of URLs. Clicking on a URL, the user can visualize a resource, and using a button in the GUI, he can associate the last semantic query to the resource being displayed. As a result, an RDF annotation is generated and inserted into the Knowledge Repository, which also inserts the annotation inside the SQAPS network in order to share it with other peers. As the only way to include an annotation is clicking on the button, the user can decide at any moment what information should be shared, minimizing privacy problems.

3.2 The SQAPS Ontology

As we have previously said, we expect that not only final human users, but also software components, could access the SQAPS network looking for semantic descriptions of Web resources. Taking this into account, we have formalized what an annotation means in the context of SQAPS system. In order to do so, we have defined a lightweight RDFS ontology, which we have represented in figure 2 in RDF N3 format. The main components of current SQAPS ontology are:

Classes The most important class in our ontology is the *SQAPSAnnotation* class. Concrete annotations of Web resources will be instances of that class. In order to be as compatible as possible with existent state of the art tools, we have defined the SQAPSAnnotation class as a subclass of the main class of the Annotea system ontology [25], *Annotation.* This way we expect that SQAPS system annotations will also be valid Annotea annotations. Apart from this class, we have defined several others, like *SQAPSAnnotatedQuery*, which is used to represent concrete annotated queries, *SQAPSPeer*, whose instances will be concrete SQAPS network nodes, *SQAPSQueryTerm*, used to represent query terms, and *SQAPSSemanticSource*, whose instances will be concrete Semantic Sources used to annotated query terms.

Properties We have defined in our ontology properties which allow to relate an SQAPSAnnotation instance with the resource being annotated (*annotates*), with the SQAPSAnnotatedQuery which represents the annotation contents (*body*), with the SQAPSPeer where the annotation has been generated (*author*), and with a timestamp which represents the instance creation time (*created*). As can be seen in figure 2, these properties are related with the correspondent ones in the Annotea ontology. Apart from these properties, we have defined others specific of the SQAPS framework, like (*term*), which allows to relate an annotated query with its terms, (*text*), used to relate a term with its string contents, (*semsource*), which represents the semantic source used to annotate a term, and (*concept*) which relates the term with its intended meaning in a source.

In figure 3 we can see an example of SQAPSAnnotation instance, which associates to the Internet resource *http://caml.inria.fr/* the query *Caml programming language* where the term *programming language* is annotated using WordNet 1.7 as Semantic Source. For space reasons we have replaced the URIs of the annotation, the semantic query, the terms in the query and the SQAPS peer, by alias. These URIs will be automatically generated by the SQAPS system, taking as basis the SQAPSPeer URI, which will be based on an automatically generated Universal Unique IDentifier, (UUID).

3.3 Prototype Implementation

In order to test the concept behind our system and its usability, we are currently working on a first basic prototype of the SQAPS system. It is being developed in Java and consists of a Swing GUI, which interacts with the Query Analysis component. This component tokenizes user queries and sends the tokens to the WordNet 1.7 lexicon, our current Semantic Source. In order to connect WordNet with our application, we are using the Java WordNet Library [26].

Once the query is annotated, it is sent to the Query Execution component, which looks for related information in the Knowledge Repository. This repository is being currently implemented using Jena [27] as RDF(S) management system. If useful information is found, it is shown to the user, so he can decide to look for more information or not. If more information is required, the plain keywords of the user query are sent to the Google Web Service [28] to get a list of URLs which match that query.

The URLs obtained as result are shown to the user, who clicking on a button can generate an annotation for a resource. This annotation is inserted into the Knowledge Repository and into the SQAPS network, in order to be shared with other users. For the initial P2P network implementation, we plan to use JXTA 2 framework, which offers a dynamic, distributed, virtual hash table infrastructure [29]. Other possibilities, like the usage of pure DHT infrastructures as CAN [30] or Pastry [31], might be explored in the future.

```
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix annotea: <http://www.w3.org/2000/10/annotation-ns#> .
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
@prefix sqaps:   <http://www.it.uc3m.es/GAST/TechWeb/SQAPS#> .

# Class definitions

sqaps:SQAPSAnnotation rdf:type rdfs:Class  .
sqaps:SQAPSAnnotatedQuery rdf:type rdfs:Class .
sqaps:SQAPSPeer rdf:type rdfs:Class .
sqaps:SQAPSQueryTerm rdf:type rdfs:Class .
sqaps:SQAPSSemanticSource rdf:type rdfs:Class .
sqaps:SQAPSAnnotation rdfs:subClassOf annotea:Annotation  .

# Property definitions

sqaps:annotates rdf:type rdf:Property .
sqaps:annotates rdfs:subPropertyOf annotea:annotates .
sqaps:annotates rdfs:domain sqaps:SQAPSAnnotation .
sqaps:annotates rdfs:range rdf:Resource .

sqaps:body rdf:type rdf:Property .
sqaps:body rdfs:subPropertyOf annotea:body .
sqaps:body rdfs:domain sqaps:SQAPSAnnotation .
sqaps:body rdfs:range sqaps:SQAPSAnnotatedQuery .

sqaps:created rdf:type rdf:Property .
sqaps:created rdfs:subPropertyOf annotea:created .
sqaps:created rdfs:domain sqaps:SQAPSAnnotation .
sqaps:created rdfs:range xsd:dateTime .

sqaps:author rdf:type rdf:Property .
sqaps:author rdfs:subPropertyOf annotea:author .
sqaps:author rdfs:domain sqaps:SQAPSAnnotation .
sqaps:author rdfs:range sqaps:SQAPSPeer .

sqaps:term rdf:type rdf:Property .
sqaps:term rdfs:domain sqaps:SQAPSAnnotatedQuery .
sqaps:term rdfs:range sqaps:SQAPSQueryTerm .

sqaps:text rdf:type rdf:Property .
sqaps:text rdfs:domain sqaps:SQAPSQueryTerm .
sqaps:text rdfs:range xsd:string .

sqaps:semsource rdf:type rdf:Property .
sqaps:semsource rdfs:domain sqaps:SQAPSQueryTerm .
sqaps:semsource rdfs:range sqaps:SQAPSSemanticSource .

sqaps:concept rdf:type rdf:Property .
sqaps:concept rdfs:domain sqaps:SQAPSQueryTerm .
sqaps:concept rdfs:range rdf:Resource .
```

Fig. 2. The SQAPS Ontology in RDF N3 format

Of course if the results of this prototype testing are promising, we also plan in the near future to integrate our application with popular Web browsers using plug-ins or other Web browser extension technologies.

```
annotation-uri      rdf:type sqaps:SQAPSAnnotation .
annotation-uri      sqaps:annotates http://caml.inria.fr/ .
annotation-uri      sqaps:created "2002-10-10T12:00:00+05:00"^xsd:dateTime  .
annotation-uri      sqaps:author uuid-SQAPS-peer .
annotation-uri      sqaps:body semantic-query-uri .

semantic-query-uri  rdf:type sqaps:SQAPSSemanticQuery .
semantic-query-uri  sqaps:term query-term1-uri .
semantic-query-uri  sqaps:term query-term2-uri .

query-term1-uri     rdf:type sqaps:SQAPSQueryTerm .
query-term1-uri     sqaps:text "Caml"^xsd:string .

query-term2-uri     rdf:type sqaps:SQAPSQueryTerm .
query-term2-uri     sqaps:text "programming language"^xsd:string .
query-term2-uri     sqaps:semsource http://wordnet.princeton.edu/v1_7 .
query-term2-uri     sqaps:concept http://wordnet.princeton.edu/v1_7#synset05760423 .
```

Fig. 3. An example of SQAPS annotation in RDF N3 format

4 Discussion

4.1 Community formation for Knowledge sharing

In the context of SQAPS, annotations are defined taking as basis user queries. As queries typically reflect user interests, we can use such information to build user profiles. These profiles can later be exploited for formation of communities of peers with common interests. These communities will be used for knowledge sharing purposes, allowing the implementation of services like a semantic community search engine, which would provide relevant resources for a certain annotated query, or a suggestion service, which would provide to the user new interesting resources found by other community members.

In order to integrate this facility into the SQAPS system, we could exploit the authorship information associated to a certain annotation. For instance, when a user decides to annotate a resource, the SQAPS system could look for annotations of such resource introduced into the P2P network by other peers. From that annotations, the system would obtain a list of SQAPSPeer instances, representing peers which could be interested in the same topics as it is. Then, contacting that peers and using a profile correlation algorithm, the local node can decide if it is interesting to keep on connecting with that peers or not.

4.2 Some SQAPS drawbacks

The SQAPS system is currently a work in progress, and has drawbacks which should be solved in the near future, some of them are shown in this section:

User collaboration required One of the main drawbacks of our system is the requirement of user collaboration. In general, the more effort requires a system from their users, the lesser will be the number of system users. In any case, from our point of view, integrating the annotation activities with habitual user actions, as Web search, should be a point in favor. Moreover, if we have into account that queries are not too long, and that query processing is partly done by the system, we expect that the work finally done by the users will imply a low overhead compared to classical keyword-based search.

Semantic Source limitations Another problem with the current system, comes from Semantic Source derived limitations, for instance:

- We can not expect that every possible meaning of every possible query term will be available in the Semantic Source.
- The current implementation of SQAPS system uses a static Semantic Source system, WordNet 1.7. New entities appear with time and Semantic Source maintenance in a distributed P2P system could be a problematic issue.

More dynamic approaches should be studied in the future, including the possibility of allowing users to define new Semantic Source entries, mainly instances representing named entities (persons, organizations, etc), which are typically used in queries. Of course, trustness approaches are crucial in this case.

Malicious users Security and trustness aspects are not currently covered in our current prototype, and are left for future work.

5 Conclusions and Future Lines

In this paper we have introduced the SQAPS, *Semantic Query-based Annotation, P2P Sharing*, system. The main idea behind this system is to exploit keyword-based user queries in semantic annotation of Web resources. Instead of annotating directly Web resources, as most of current systems in the state of the art suggest, we have proposed a system in which users annotate their queries. In order to share these keyword-based semantic annotations with other SQAPS users, a semantic P2P infrastructure is used.

From our point of view, this alternative approach to annotation of Web resources has the advantage that could exploit the force of the millions of users which every day look for information on the Web. Additionally it supports the annotation of different kinds of resources. This is so because in the context of the SQAPS system, an annotation is an association of a certain URL, representing a Web resource, and a semantic query. As resources can be multimedia files and not only text or HTML, in principle the annotation of multimedia items is supported, but, of course, only in the case that those items can be retrieved using a keyword-based query and a classical Web search engine.

Apart from finishing the prototype, testing its viability, and look for concrete solutions for the drawbacks introduced in previous section, future lines of development of SQAPS system could include the following:

Multilingual annotations In the prototype of SQAPS that we are currently developing, we are using the WordNet English lexicon to analyze user queries and add semantic to them. A possibility that we want to explore in the near future is the usage of a multilingual lexicon like EuroWordNet [32] instead of WordNet. With this substitution, and without changing the SQAPS architecture, we could provide a multilingual annotation tool.

Community formation As we have argued, the information in the SQAPS network could be helpful for P2P community formation. This functionality is not included in the prototype being implemented, but we expect to add it in the near future.

Acknowledgements

This work has been partially funded by the *Ministerio de Educación y Ciencia de España*, as part of the Infoflex Project, TIC2003-07208.

References

1. S. Handschuh, S. Staab (Eds.), "Annotation for the Semantic Web". Ed. IOS Press, ISBN 1-58603-345-X, 2003.
2. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004. Available at: http://www.w3.org/TR/rdf-schema/
3. Resource Description Framework (RDF). Available at: http://www.w3.org/RDF/
4. Nejdl, W.; Wolf, B.; Qu, C.; Decker, S.; Sintek, M.; Naeve, A.; Nilsson, M.; Palmér, M.; Risch, T.; EDUTELLA: A P2P Networking Infrastructure Based on RDF. In 11th International World Wide Web Conference, WWW2002, Honolulu, Hawaii, USA, May 2002.
5. Cai, M.; Frank, M.; RDFPeers: A Scalable Distributed RDF Repository based on a Structured Peer-to-Peer Network. In 13th International World Wide Web Conference, WWW2004, New York, USA, May 2004.
6. G. Tummarello, C. Morbidoni, J. Petersson, P. Puliti, F. Piazza. RDFGrowth, a P2P annotation exchange algorithm for scalable Semantic Web applications. In 2nd International Workshop on Peer-to-Peer Knowledge Management, P2PKM04.
7. Mukherjee, S.; Yang, G.; Ramakrishnan, I.V.; Automatic Annotation of Content-Rich HTML Documents: Structural and Semantic Analysis. The Semantic Web - ISWC 2003, LNCS 2870, pp 533-549.
8. Dill, S.; Eiron, N.; Gibson, D.; Gruhl, D.; Guha, R.; Jhingran, A.; Kanungo, T.; Rajagopalan, S.; Tomkins, A.; Tomlin, J.A.; Zien, J.Y.; SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. WWW2003 Conference, May 2003, Budapest, Hungary.
9. TAP: Building the Semantic Web. Available at: http://tap.stanford.edu/
10. Kahan, J.; Koivunen, M-R.; Prud'Hommeaux, E.; Swick, R.R.; Annotea: An Open RDF Infraestructure for Shared Web Annotations. In WWW10 Conference, May 1-5 2001, Hong Kong.
11. Kogut, P.; Holmes, W.; AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages. First International Conference on Knowledge Capture (K-CAP 2001). Workshop on Knowledge Markup and Semantic Annotation, Victoria, B.C. October 21, 2001.
12. Handschuh, S.; Staab, S.; Authoring and Annotation of Web Pages in CREAM. Proceedings of the 11th International World Wide Web Conference, WWW 2002, Honolulu, Hawaii, May 7-11, 2002. ACM Press.

13. Handschuh, S.; Staab, S.; Ciravegna, F.; S-CREAM: Semi-automatic CREAtion of Metadata Proceedings of the European Conference on Knowledge Acquisition and Management - EKAW-2002. Madrid, Spain, October 1-4, 2002.
14. Cimiano, P.; Handschuh, S.; Staab, S.; Towards the Self-annotating Web. In the 13th International World Wide Web Conference, WWW 2004, New York, USA, May 17-22, 2004
15. Google Web Search Engine. Available at: http://www.google.com
16. Kalyanpur, A.; Hendler, J.; Parsia, B.; Golbeck, J.; SMORE - Semantic Markup, Ontology, and RDF Editor. Available at: http://www.mindswap.org/papers/SMORE.pdf
17. Bechhofer, S.; Goble, C.; Carr, L.; Kampa, S.; COHSE: Semantic Web gives a Better Deal for the Whole Web? Poster presentation at ISWC International Semantic Web Conference, Sardinia, June 2002.
18. Vargas-Vera, M.; Motta, E.; Domingue, J.; Lanzoni, M.; Stutt, F.; Ciravegna, F.; MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In 13th International Conference on Knowledge Engineering and Management (EKAW 2002), Springer Verlag, 2002.
19. SHOE Knowledge Annotator. Available at: http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html
20. SHOE: Simple HTML Ontology Extensions. Available at: http://www.cs.umd.edu/projects/plus/SHOE/index.html
21. Handschuh, S.; Staab, S.; Volz, R.; On Deep Annotation. In 12th International World Wide Web Conference, WWW2003, Budapest, Hungary, May 2003.
22. Harren, M.; Hellerstein, J.M.; Huebsch, R.; Loo, B.T.; Shenker, S.; Stoica, I.; Complex Queries in DHT-based Peer-to-Peer Networks. Available at: http://www.cs.rice.edu/Conferences/IPTPS02/191.pdf
23. Li, J.; Loo, B.T.; J.M. Hellerstein, J.M.; Kaashoek, M.F.; On the Feasibility of Peer-to-Peer Web Indexing and Search. In 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), February 2003, Berkeley, CA, USA.
24. WordNet, a lexical database for the English language. Available at: http://wordnet.princeton.edu/
25. Annotea annotation schema. Available at: http://www.w3.org/2000/10/annotation-ns
26. SourceForge.net: Project Info - JWNL (Java WordNet Library) Available at: http://sourceforge.net/projects/jwordnet
27. Jena - A Semantic Web Framework for Java. Available at: http://jena.sourceforge.net/index.html
28. Google Web APIs Home. Available at: http://www.google.com/apis/
29. JXTA 2: A high-performance, massively scalable P2P network. Available at: http://www-106.ibm.com/developerworks/java/library/j-jxta2/
30. Ratnasamy, S.; Francis, P.; Handley, M.; Karp, R.; A Scalable Content-Addressable Network. In Proceedings ACM SIGCOMM'01, pp. 161-172, August 27-31, 2001, San Diego, CA, USA.
31. Rowstron, A.; Druschel, P.; Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001). Heidelberg, Germany, November 2001.
32. EuroWordNet. Building a multilingual database with wordnets for several European languages. Available at: http://www.illc.uva.nl/EuroWordNet/