# Searching Community-built Semantic Web Resources to Support Personal Media Annotation

Annett Mitschick, Ronny Winkler, and Klaus Meißner

Dresden University of Technology, Department of Computer Science
Chair of Multimedia Technology, 01062 Dresden, Germany
{annett.mitschick, ronny.winkler, klaus.meissner}@inf.tu-dresden.de

**Abstract.** Appropriate annotation of documents is a central aspect of efficient media management and retrieval. As ontology-based description of documents and facts enables exchange and reuse of metadata among communities and across applications, the annotation of personal media collections using Semantic Web technologies benefits from existing (and evolving) information sources on the Internet. This paper addresses conceptual and technical issues of Web search within community-built Semantic Web content to retrieve useful information for personal media annotation. After analyzing application scenarios, we introduce a generic and extensible Semantic Web Search Component, which facilitates specific search configurations. As a sample application, we deployed the component within our ontology-based media management system, including evaluation and remarks on quantity and quality of search results with regard to community-built Semantic Web content.

## 1 Introduction

Personal media collections comprehend knowledge representing context and individual view of the owner. This knowledge is the ultimate key for managing digital media collections in a way that is suitable for human beings. However, to enable applications to process and visualize navigation paths and arrangements based on people's knowledge, appropriate machine-processable descriptions are needed. Semantic Web technologies [1], [2] provide opportunities to create and share such ontology-based descriptions in a standardized way.

The question which arises is how those semantic descriptions could be generated or reused from existing information sources. To some extend information about digital documents exists explicitly in the form of annotations and metadata (for specific formats quite comprehensive and according to established standards like ID3, EXIF, IPTC, XMP, etc.). On the other hand, a substantial portion of knowledge results implicitly from the content itself (e.g. persons or locations depicted in a photograph), the structure and characteristic features of a document, etc. Machine Learning techniques and classification might solve the one or the other issue. However, the safest way to acquire semantic information is to ask the user to make contributions. With regard to the user's comfort, manual annotation should be limited to the most necessary, reusing existing information on the local desktop or even on the World Wide Web.

In this paper we propose an approach to enhance annotation of multimedia documents using semantic resource descriptions and ontology models found on the WWW. The first part of Section 2 illustrates background and state-of-the-art of ontology-based media management and media annotation, referencing relevant related work. The second part addresses opportunities to search the Semantic Web (in particular RDF-based user/community generated content) using Web Services and crawler implementations. We also present a selection of use cases of media annotation supported by Semantic Web search. Section 3 introduces our Semantic Web Search Component, illustrated and evaluated in Section 4 using a sample application. Finally, conclusion and outline of future work is given in Section 5.

## 2 Ontology-based Annotation of Documents

Documents are or can be enriched with metadata and annotations in several ways and on several levels. Chakravarthy et. al. [3] introduced five "dimensions" of information associated with documents: *resource metadata* (e.g. creation date, author, etc.), *content annotation* (describing information within the document), *immutable knowledge* (e.g. knowledge from dictionaries), *informal knowledge* (e.g. knowledge not explicitly mentioned within the document), and *folksonomies* (cf. Flickr[1] or Del.icio.us[2]). Following this classification, a couple of solutions and applications for document annotation address the one or the other "level", depending on the used ontologies and concepts. However, most of the work on ontology-based annotation (like CREAM [4], AKTive Media [3]) proceeds from the assumption that, before annotation, an appropriate ontology has to be created or assigned as a description schema (top-down approach). If this is left to the user, modality and sense of annotations depend on his/her intension, which is even more difficult for non-ontology engineers.

Even if a lot of projects are dedicated to general "cross-media" annotation, regarding supported application scenarios they either focus more on text resp. Web content annotation (like Annotea [5]) or multi-media annotation (like M-OntoMat-Annotizer [6]). In order to ease the manual effort of annotation, several projects apply Information Extraction and Machine Learning techniques to populate descriptions (in particular Natural Language Processing in case of text documents). However, as annotations can hardly be automated completely (regarding subjective information, fuzzy knowledge, etc.) the user should be encouraged to make individual contributions. In this regard, aspects of community contributions and "social annotation" offer interesting opportunities. Bookmark and tagging services enjoy growing popularity as their success particularly bases on the low entry barrier [7].

[1] `http://www.flickr.com`
[2] `http://del.icio.us`

Ontology-based annotation of private media collections could profit from recent developments, not only restricted to the incorporation of folksonomies, but in general through information sources from the current Semantic Web, which will likely evolve with the help of the Web community and appropriate applications.

## 2.1 Finding Semantic Web Content on the WWW

Dedicated Semantic Web search engines facilitate a focused access to Semantic Web content. Their operating mode is similar to traditional Web search engines. Thus, a Semantic Web search engine also consists of a crawler ("robot" or "spider"), a database, and a search interface.

Crawling the Semantic Web is comparable to crawling the Web of HTML content [8]. A crawler starts with some seed URLs, downloads the corresponding documents, analyzes each document to gather further URLs for crawling and does context specific processing of the retrieved contents, like creating the searchable entries in the database. The last steps are repeated until a stop criterion is met (e.g. no more URLs to crawl, reached a predefined link depth, or gathered a predefined amount of documents). In case of a Semantic Web crawler the documents of relevance are those containing RDF-based data, and the goal of discovering unvisited URLs from previously retrieved RDF data can be achieved through evaluating statements with predicates which are capable of expressing relationships between documents, like *rdfs:seeAlso* or *owl:imports*.

There are several standalone Semantic Web crawler implementations, which can be used for purpose-built search engines or software projects. Some to mention here are the crawler of the KAON framework [9], the Slug crawler [10], and RDF-Scutter[3]. However, using a standalone crawler, exhaustive crawling is needed to create a passably extensive database of Semantic Web data. This requires considerable amounts of time, disk space, and Web transfers for collecting and maintaining the data. Therefore, available search services like Swoogle [11], which offers support for software agents via a REST interface [12], can be used more easily in an application to profit from rich databases. Currently, Swoogle has parsed and indexed more than 370 million triples from about two million Semantic Web documents[4]. It allows search for terms, documents, and ontologies (i.e. a subset of Semantic Web documents where the fraction of defined classes or properties is significantly higher than the fraction of instances). A Swoogle query is basically a set of keywords which should be found in the literal descriptions of indexed documents, terms, or in the URIrefs of defined classes or properties. A query initiated by a software agent is responded with an RDF/XML file containing the ranked search results. Testing Swoogle showed that its strength is more on the side of finding ontologies, than of finding documents with instance data. Nevertheless, the major drawback of using a remote search engine within applications is of course the dependency on its availability and maintenance, which should be taken into account.

[3] `http://search.cpan.org/src/KJETILK/RDF-Scutter-0.1/README`
[4] `http://swoogle.umbc.edu/index.php?option=com_swoogle_stats`

## 2.2 Using Semantic Web Content for the Annotation of Personal Media Collections

A variety of media analyzing and information extraction tools (e.g. [6]) are able to perform the task of extracting characteristic attributes and features (including inherent metadata) from media documents for the generation of semantic descriptions. As already mentioned, automatically extracted information might not be sufficient enough for an appropriate description. In the following, we give some conceivable use cases for the further refinement of basic, automatically generated information with the help of external resources, in particular retrieved by the Semantic Web search component, introduced in Section 3:

*Assigning terms or categories from a glossary or thesaurus:* The user wants to add a tag to a document to assign it to a category or concept. Actually, he is not sure about the proper term and wants to use existing definitions (perhaps a controlled vocabulary). He discovers a domain thesaurus on the Web (e.g. a SKOS [13] based document), which contains suitable items to assign to the document.

*Referring to domain-specific descriptions of people, social events, communities, projects, etc.:* Analyzing components may extract - among others - the name (family and given name) of the photographer from the metadata of an image. A resource description of this person was generated but without further information than the name literals. Searching the Semantic Web possibly returns a *Friend-of-a-Friend* (FOAF) or vCard description of the person (or one with a similar name). The user can decide to add the found resources to his model to extend the description of the photographer. Moreover, some documents might be related to resources, like events (e.g. a party, workshop, trip, etc.) or work projects. In addition to his personal view and context, the user might want to link to external descriptions maintained by a community.

*Referring to a Web page with embedded RDF:* Besides Semantic Web documents containing pure RDF resp. OWL data, RDFa [14] annotated XHTML documents could as well provide relevant resource descriptions, e.g. published events, contacts, etc. In addition to the previous use case, the user might want to keep the link to the annotated Web page containing the retrieved information.

*Adopting domain specific description schemes:* The basic ontology model might not be sufficient enough to describe special issues, regarding diverse interests, profession, and background of users (e.g. detailed interest in wine, zoology, classical music, etc.). A keyword-based search might lead the user to an appropriate ontology on the Internet which he could adopt.

*Improve information extraction from text documents:* The results of *Named-Entity-Recognition* (NER) in text documents could be qualified by semantic search results, i.e. tagging person names, addresses, locations, events, etc. within the document depending on found entities on the Semantic Web.

According to these and other identified application scenarios, we finally derived the following concepts of information reuse within the context of annotation, each with increasing complexity:

***Tagging:*** Assigning tags to multimedia content (or generally any resource in the model) is probably the easiest way of information integration and does not necessarily require substantial adjustment of the ontology model. The RDF vocabulary [15] provides built-in utility properties for linking between general resources. One of those is *rdf:seeAlso*, which could be used as a simple tag relation between two *rdfs:Resource* instances. A better representation of the semantics of tagging might certainly be the definition of a tagging vocabulary (*hasTag*, *taggedBy*, etc.) to combine benefits of a controlled vocabulary with those of social tagging.

***Referencing external objects:*** Found resources on the Web might be integrated as objects of a defined property if they fit in the required range, i.e. the same class or subclass. The practical application of this option depends on the constraints within the used ontology model. Proprietary object types of course complicate the creation of semantic nets to external resources.

***Instance mapping:*** In the case of instance mapping, attributes and data of the retrieved resource are "translated" to slots of the target resource. Therefore, adjustment of the ontology model is not needed. Hints how to solve concrete mapping problems should be given by the user.

***Refinement (specialization):*** A specialization of classes within the ontology model using retrieved class definitions or class definitions of retrieved instances might be useful. In the concrete application scenario the user introduces this subclass relation with a retrieved instance. He wants the target instance to adopt its properties, but keep the existing class definition unaffected. The retrieved class is incorporated into the ontology model as a copy and defined as subclass of the target class. The target instance is altered to an instance of the new class. Thus, existing relations to the instance are still valid.

***Instance and schema adoption:*** The most complex scenario of information reuse from retrieved resources is the extension of the ontology model with both instances and their according schema. The user wants to incorporate a resource as object of a newly defined property of an existing resource. Thus, the ontology model has to be extended with the new property and a local copy of the adopted class.

Please note, that all of these concepts refer to crawled data in general, which could be downloaded from the Internet to local disk or used without local caching. Thus, as models, once retrieved, could change or get lost, cached data becomes obsolete, but without caching statements might become invalid. Therefore, its left to the developer to find a reasonable compromise.

## 3 Semantic Web Search Component (SWSC)

Based on the study of existing Semantic Web search solutions and use cases (see Sections 2.1 and 2.2) we developed a Semantic Web Search Component (SWSC), as depicted in Fig. 1. The SWSC is designed to extend applications of Semantic Web technologies with search functionality, including search for ontologies, documents with instance data, and terms. Instead of creating our own crawling infrastructure, we decided to reuse existing Web search services in the form of meta-crawling. As it seemed advisable to reduce the dependency on a single service, we provide an extensible meta-crawler concept (cf. Fig. 1), facilitating dedicated *Crawler* implementations handled by a central *CrawlerManager*. The main idea of this approach is a generic interface (*WebSearchInterface*) which accepts search requests and forwards them to the registered crawler implementations. A more detailed description of the search requests is given below in Section 3.1.

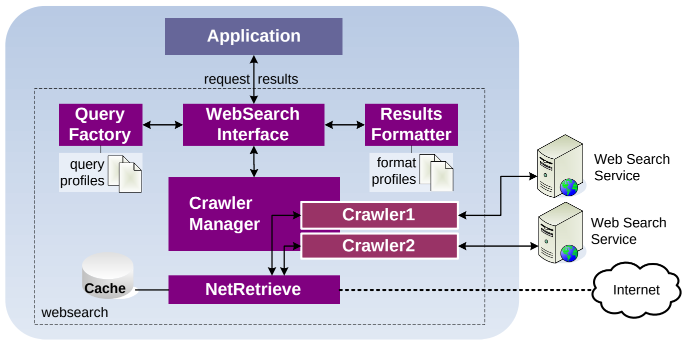


**Fig. 1.** Search concept and integration of the SWSC.

After processing the search task, each crawler implementation produces an initial result set of potential document URIs found on the WWW (i.e. indexed by the inquired Web service), which are evaluated in the following according to the given search criteria. To achieve the required Web communication, the *NetRetrieve*-component, offering multi-threaded downloads with local caching capabilities, was implemented. The *CrawlerManager* removes duplicate results, if documents have been found by several *Crawlers*, and applies the predefined filters of the *Query* object to the result set. The *ResultsFormatter* finally generates an adequate representation of the search results to be returned, according to specified "format profiles"(i.e. templates for XML, XHTML, or RDF response).

Trying to harvest community-built Semantic Web content, we decided to combine a dedicated and a general purpose search engine to achieve a better coverage, regarding the identification of potential instance data. Currently our SWSC implementation makes use of the publicly available Web interface of the

Semantic Web search engine Swoogle in combination with the general purpose Web search service of Yahoo!. As a matter of course, Swoogle's strength is the dedicated and exclusive access to Semantic Web content (ontologies, documents, and terms) which has already been evaluated and ranked. As mentioned before in Section 2.1, Swoogle's ability to supply instance data is relatively limited. Although the coverage of Yahoo! is estimated to be smaller than that of Google[5], we decided to work with Yahoo! as Swoogle itself already applies Google-based meta-crawling for its index [16].

## 3.1 Defining Search Requests and Results Filtering

Needless to say, requests to the Web interfaces of the search engines have to conform to the required query syntax and parameters. The implemented *Crawlers* serve as wrappers for the request specification and reception of results from the according Web interface. A general *Query* object is used to retain the implementation-independent query parameters and filter definition. The query string itself consists of keywords to be searched for. Additionally, different kinds of filters could be attached to a query to constrain the search. All these filters can be operated as blacklists or whitelists, allowing conjunction or disjunction. Currently, we use a host filter to include or exclude results from specific hosts, as well as a file type filter to restrict the result set of documents, a namespace filter which can be used to test if an RDF-model relies on a specific vocabulary, and a filter that checks terms whether they fulfill certain criteria (if they are class or property, or subject/objects of a specific statement). In general, the filters are applied in this order. Although not all of them can be mapped directly to the query syntax of the Web search interfaces[6], they are used to evaluate the retrieved documents locally to determine in more detail whether they match the query.

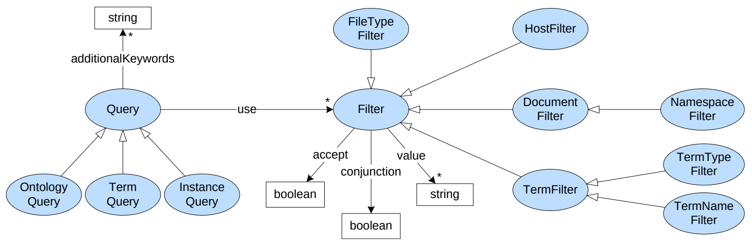


**Fig. 2.** Query profile ontology. For each filter (combined by conjunction) a set of restricting values (e.g. URLs) can be specified and combined by disjunction or conjunction (*disjunction=true/false*), and as whitelists or blacklists (*accept=true/false*).

[5] `http://blog.searchenginewatch.com/blog/050517-075657`, May 2005

[6] e.g. using file type restriction within the Web query with *url:* (Swoogle) resp. *originurlextension:* (Yahoo!)

Based on the requirements for a general query definition we created a purpose-built ontology for the Web query specification (an extract is given in Fig. 2). Thus, we are able to instantiate some sort of "query profiles" as instances for specific application scenarios (people search, schema search, etc.), which can be loaded by the *QueryFactory* to instantiate the appropriate *Query* object.

To prove applicability of Semantic Web search within the context of personal media annotation, we integrated the SWSC into our ontology-based media management system, which we illustrate in the following.

## 4 Semantic Web Search within the *K-IMM* Ontology-based Media Management System

This work is based on the results and implementation within the *K-IMM* (Knowledge through Intelligent Media Management) project, which provides a system architecture for intelligent media management for private users (i.e. semi- or non-professionals) [17]. The intention of this project is to take advantage of ontology engineering and Semantic Web technologies in such a way that users without particular skills can interact intuitively and without additional cost. Therefore, the system comprises components for automated import and indexing of media items (of different type) as background tasks [18]. A conceptual overview of the overall architecture of the *K-IMM* System is depicted in Figure 3.

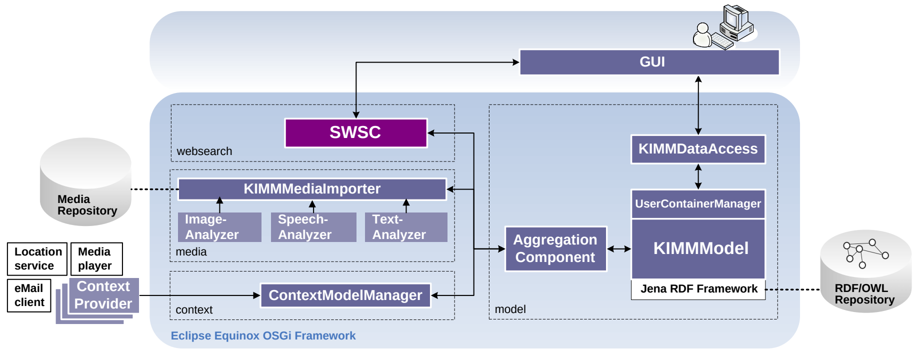


**Fig. 3.** The overall *K-IMM* System architecture.

All of the components are realized as plug-ins (bundles), according to the OSGi specification [19], and developed and run within the Eclipse Equinox execution framework. Thus, further plug-ins for specific media type analysis and processing or advanced components for visualization can easily be added to the system, and can be started and stopped dynamically at runtime. Hence, it is possible to run the system e.g. just for image management (starting only image analyzing and image semantics deducing components), or only with low-level indexing (without semantic modeling), if desired. The media analyzing components extract available properties and features and pass them to the knowledge modeling components. In our prototype implementation RDF and OWL processing, storage and reasoning is based on the Jena Framework[7] including the Jena Inference Support. Further components, which are also not subject of this paper, comprise context aggregation and modeling.

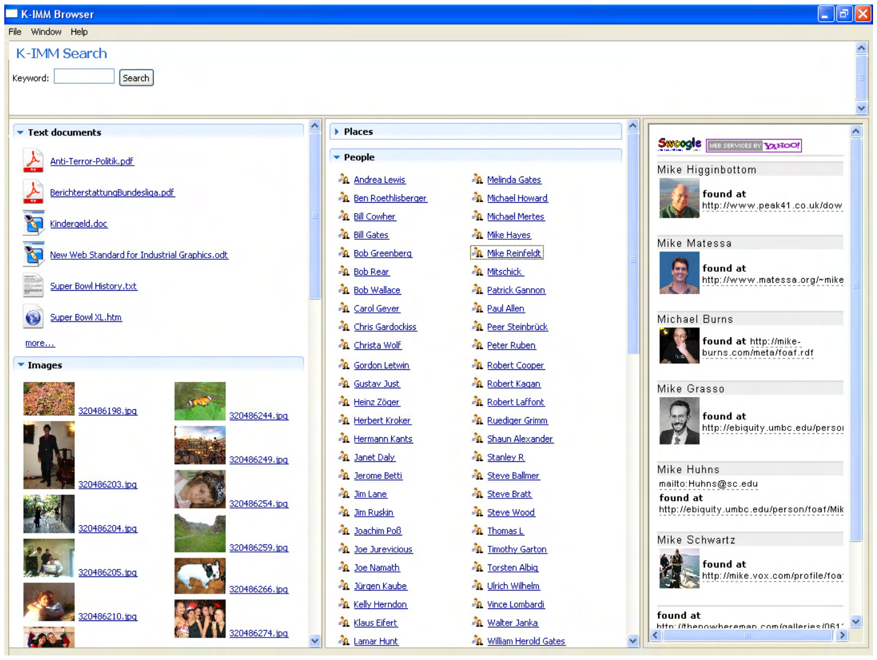
## 4.1 Example: People Search

We set up a purpose-built graphical user interface (presented in Fig. 4) which shows the collection of documents (in this case images and text documents) managed by the underlying K-IMM System on the left, and extracted semantic entities (class instances based on our media management ontology) in the middle. In this example, the semantic entities were generated from people's names and locations, detected in the text documents using Named-Entity-Recognition methods and in metadata of the digital photographs. While the test set of text documents have been collected from the Internet and local desktop, the pictures were recent uploads at Flickr we downloaded via Flickr Web API. Thus, we could obtain Flickr users' names (i.e. first names, family names, and nicknames), EXIF metadata, and in some cases also location information from "geo-tagged" photos.

The semantic entities are represented in a categorized list. Clicking on the person entries starts the type specific Web search. Requests to the Web services were associated by default with the restriction to the appropriate file types (*FileTypeFilter*) and hosts (*HostFilter*) to limit the size of the initial result set. Additionally, we restricted the retrieved results from Swoogle even more, passing namespace filtering parameters (*ns:foaf*, etc.), which is of course not possible with Yahoo!. If exact match of the search phrase fails in a first try (initial query to the engine), the SWSC automatically retries the keywords with logical disjunction, to broaden the initial result set a bit, and leave further filtering to the document and term filters after download. We learned that this approach worked best in our case, although in most cases search results are "only" similar to the person we searched for (cf. Fig. 4: in this example we searched for "Mike Reinfeldt". After exact match failed, the SWSC broadens search to find similar names, in this case resulting in a set of other "Mikes").

The potential search results are represented in a list on the right side. Their representation is generated by the *ResultsFormatter* (mentioned in Section 3), which in this example returns a type-specific XHTML representation of the found people descriptions (mostly FOAF documents, their possibly contained *foaf:depiction* entries are used here to give a visual representation of the person). Clicking on other types of entities (e.g. a place) passes other types of search request to the SWSC, resulting in a specific *Query* object with according filter configuration (e.g. searching within address fields of vCard documents).

[7] `http://jena.sourceforge.net/`



Anti-Terror-Politik.pdf
BerichterstattungBundesliga.pdf
Kindergeld.doc
New Web Standard for Industrial Graphics.odt
Super Bowl History.txt
Super Bowl XL.htm

320486198.jpg
320486244.jpg
320486203.jpg
320486249.jpg
320486254.jpg
320486204.jpg
320486259.jpg
320486205.jpg
320486266.jpg
320486210.jpg
320486274.jpg

| | |
|---|---|
| Andrea Lewis | Melinda Gates |
| Ben Roethlisberger | Michael Howard |
| Bill Cowher | Michael Mertes |
| Bill Gates | Mike Hayes |
| Bob Greenberg | Mike Reinfeldt |
| Bob Rear | Mitschick |
| Bob Wallace | Patrick Gannon |
| Carol Geyer | Paul Allen |
| Chris Gardockiss | Peer Steinbrück |
| Christa Wolf | Peter Ruben |
| Gordon Letwin | Robert Cooper |
| Gustav Just | Robert Kagan |
| Heinz Zöger | Robert Laffont |
| Herbert Kroker | Ruediger Grimm |
| Hermann Kants | Shaun Alexander |
| Janet Daly | Stanley R |
| Jerome Betti | Steve Ballmer |
| Jim Lane | Steve Bratt |
| Jim Ruskin | Steve Wood |
| Joachim Poß | Thomas L |
| Joe Jurevicious | Timothy Garton |
| Joe Namath | Torsten Albig |
| Jürgen Kaube | Ulrich Wilhelm |
| Kelly Herndon | Vince Lombardi |
| Klaus Eifert | Walter Janka |
| Lamar Hunt | William Herold Gates |

Mike Higginbottom
**found at** http://www.peak41.co.uk/dow

Mike Matessa
**found at** http://www.matessa.org/~mike

Michael Burns
**found at** http://mike-burns.com/meta/foaf.rdf

Mike Grasso
**found at** http://ebiquity.umbc.edu/persoı

Mike Huhns
mailto:Huhns@sc.edu
**found at** http://ebiquity.umbc.edu/person/foaf/Mik

Mike Schwartz
**found at** http://mike.vox.com/profile/foa

**found at**

**Fig. 4.** Screenshots of the test search prototype, showing the results of a search example on the right.

## 4.2 Evaluation

Regarding the approximate usage of Semantic Web documents[8], *foaf* (`http://xmlns.com/foaf/0.1/`), *vcard* (`http://www.w3.org/2001/vcard-rdf/3.0#`), and *bio* (`http://purl.org/vocab/bio/0.1/`) are probably the most promising namespaces to find instance data describing people. Thus, people search was configured with an according *NamespaceFilter* white list and a collection of *TermNameFilter*s to evaluate whether found resources correspond to a person description ("name", "given name", "nick", "surname", etc.). To test and refine the settings of our query profile we ran a series of queries in batch mode based on a list of named entities (500 person names, i.e. first and last names), originally used for Named-Entity-Recognition. In doing so, we logged the number of initial, blocked, and accepted results, as well as the cause of the rejection, to get an idea of the quantity and quality of Semantic Web search with our implementation.

As to be seen in Fig. 5, aside from a few outliers, Swoogle returns - on average - a smaller initial result set, but with an overall higher value (less parse exceptions). Results from Yahoo! are more often blocked because of invalid content (non-RDF data). In general, document (namespace) and term filters restrict the result sets in both cases the most, as found search strings very often occur in

[8] `http://ebiquity.umbc.edu/blogger/100-most-common-rdf-namespaces/`

non-specific comments or labels not related to people descriptions. The analysis results are certainly quite evident, as Swoogle is much more dedicated to Semantic Web documents and uses a combination of Google meta-crawling, bounded HTML crawling, and RDF crawling [16]. On the other hand, we observed that in some cases Yahoo! retrieved documents which Swoogle did not find. Please note, that the application of less restrictive filters directly increases the number of accepted search results. That means, a quite high percentage of documents was actually usable Semantic Web content (available and valid RDF-based data), but blocked due to namespace or term filters for this special application scenario.

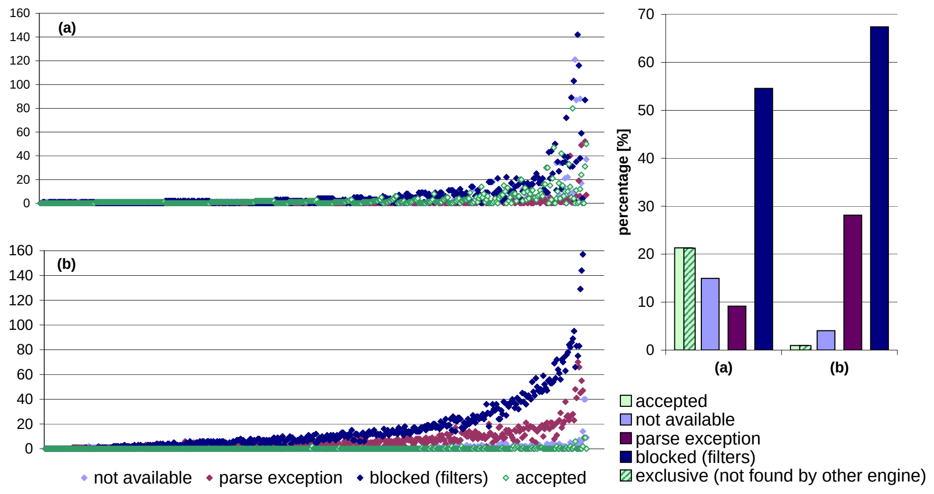


**Fig. 5.** An analysis of the results retrieved from Swoogle (a) and Yahoo! (b). On the left: absolute quantity (based on 500 people queries in March 2007, sorted along the x-axis with increasing number of initial results). On the right: relative distribution of the results quality. The *exclusive* results show, that the accepted result sets are almost disjoint.

However, there is of course a difference between tests (random lists of common named entities) and a real-world scenario of personal media annotation. People's social context is very individual, but in generally also more networked and interlinked (contacts and relationships). Thus, a general search within the range of the WWW would often fail (esp. regarding language difference). Instead, dedicated connections to community platforms (exploiting social networks), adjusted host filters, and specialized crawler implementations (e.g. using dictionaries for synonyms or different notations) should be used - which can be done within our SWSC.

In fact, today's WWW is still sparsely populated with Semantic Web content. Hence, search results are often not as expected. On the other hand, current results are quite promising and show that user generated Semantic Web content (pushed by RDF-enabled community portals) is already retrievable and applicable. With a growing amount of Semantic Web content the developed SWSC can be configured to do more sophisticated filtering and ranking of obtained search results, e.g. using combined *TermName-* and *TermTypeFilters* to reduce false positives.

## 5 Conclusion and Future Work

In this paper we discussed Semantic Web search opportunities and their benefits within the context of the annotation of personal media collections. For that purpose we identified use cases of information search and integration, and developed a Semantic Web Search Component (SWSC) as a generic plug-in for various applications, using a combination of Web search engines for meta-crawling. As a particular usage scenario we presented a people search application with graphical user interface, based on our K-IMM media management system. Finally, we evaluated the search results of the implementation to show the particular benefits of this approach.

Our general approach allows further integration and extension of crawling implementations (e.g. to harvest community portals) for various scenarios and requirements with the help of customized query profiles and formatting rules. The current difficulty of our approach is basically the lack of valid and rich Semantic Web content indexed by available Web search engines. However, our component is capable of further application-specific refinements to use specialized or purpose-built Web services in combination, to extend the coverage of Semantic Web search. Furthermore, we think that Semantic Web content will increase in the next years with the help of communities and appropriate applications. Thus, our search component will support people and applications in discovering useful information resources in a growing Semantic Web.

Target of our future work will be the implementation of alternative connections and interfaces to other search engines or information sources to broaden the potential search results. As our evaluation shows, people search would certainly work much better with a dedicated FOAF search engine which collects FOAF data following *foaf:knows* links. Moreover, we are about to extend the developed component to realize different scenarios of information reuse, as we described in Section 2.2, e.g. search for public events, conferences, etc. We will also test proactive search scenarios and their benefits to users, which yet necessitates an acceleration of the evaluation and formatting of search results. Therefore, our current prototype already stores - in addition to the mentioned caching mechanism - lists of accepted documents which have been retrieved and evaluated beforehand in a background task.

## References

1. J. J. Carroll and G. Klyne. Resource description framework (RDF): Concepts and abstract syntax. W3C recommendation, W3C, February 2004.

http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.
2. D. McGuinness and F. van Harmelen. 2004 OWL web ontology language overview. W3c recommendation, W3C, 2004. http://www.w3.org/TR/owl-features/.
3. A. Chakravarthy, F. Ciravegna, and V. Lanfranchi. Cross-media document annotation and enrichment. In *Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW2006)*, 2006. http://www.dcs.shef.ac.uk/ ajay/publications/paper-camera-workshop.pdf.
4. S. Handschuh and S. Staab. Authoring and annotation of web pages in cream. In *Proceedings of the Eleventh International World Wide Web Conference, WWW2002*, pages 462–473, 2002.
5. J. Kahan, M.R. Koivunen, E. Prud'hommeaux, and R.R. Swick. Annotea: an open rdf infrastructure for shared web annotations. In *Proceedings of the 10th International World Wide Web Conference*, pages 623–632, 2001.
6. S. Bloehdorn, K. Petridis, C. Saathoff, N. Simou, V. Tzouvaras, Y.S. Avrithis, S. Handschuh, I. Kompatsiaris, S. Staab, and M.G. Strintzis. Semantic annotation of images and videos for multimedia analysis. In *ESWC*, pages 592–607, 2005.
7. A. Mathes. Folksonomies - cooperative classification and communication through shared metadata, December 2004.
8. M. Biddulph. Crawling the semantic web. In *Proceedings of XML Europe 2004*, 2004.
9. E. Bozsak, M. Ehrig, S. Handschuh, A. Hotho, A. Maedche, B. Motik, D. Oberle, C. Schmitz, S. Staab, L. Stojanovic, N. Stojanovic, R. Studer, G. Stumme, Y. Sure, J. Tane, R. Volz, and V. Zacharias. Kaon - towards a large scale semantic web. In *EC-Web 2002, Aix-en-Provence, France, 2002, Proceedings*, volume 2455 of *LNCS*, pages 304–313. Springer, 2002.
10. L. Dodds. Slug: A semantic web crawler. In *Proceedings of Jena User Conference 2006*, 2006.
11. T. Finin, L. Ding, R. Pan, A. Joshi, P. Kolari, A. Java, and Y. Peng. Swoogle: Searching for knowledge on the Semantic Web. In *AAAI 05 (intelligent systems demo)*, July 2005.
12. R. T. Fielding. *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine, 2000.
13. A. Miles and D. Brickley. SKOS Core Guide. W3C Working Draft, World Wide Web Consortium, November 2005.
14. B. Adida and M. Birbeck. RDFa primer 1.0. W3C Editors' Draft, W3C, 2006. http://www.w3.org/2006/07/SWD/RDFa/primer/.
15. D. Brickley and R. V. Guha. RDF vocabulary description language 1.0: RDF schema. W3C recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/.
16. L. Ding and T. Finin. Characterizing the semantic web on the web. In *Proceedings of the 5th International Semantic Web Conference*, November 2006.
17. A. Mitschick. Ontology-based management of private multimedia collections: Meeting the demands of home users. In *6th International Conference on Knowledge Management (I-KNOW'06), Special Track on Advanced Semantic Technologies*, Graz, Austria, 9 2006.
18. A. Mitschick and K. Meiß ner. A stepwise modeling approach for individual media semantics. In *GI-Edition Lecture Notes in Informatics (LNI)*, Dresden, Germany, 10 2006.
19. D. Marples and P. Kriens. The open services gateway initiative: An introductory overview., 2001.