

Using Your Desktop as Personal Digital Library

Stefania Ghita

L3S Research Center / University of Hanover
Deutscher Pavillon, Expo Plaza 1
30539 Hanover, Germany
ghita@l3s.de

Abstract. The recently arrived desktop search applications are weaker than their web siblings as they cannot rely on PageRank-like ranking methods which have revolutionized web search, since the documents are not well connected on the desktop. The general aim of this thesis proposal is to discuss how to enhance and contextualize desktop search based on semantic metadata collected from different contexts (email, files and browser cache) and activities performed on a computer. We describe the semantics of these different contexts by appropriate ontologies and show how to extract and represent the corresponding context information as RDF metadata and how to use them in the context of the semantic desktop.

1 Introduction & Architecture

With the continuous growth of data stored on one's computer, it has become quite obvious that we need desktop search applications, and not the ones that do a simple indexing of the data on the desktop, but enhanced with ranking techniques, to put order into the search results. The main problem with ranking on the desktop comes from the lack of links between documents. A semantic desktop offers the missing ingredients: by gathering semantic information from user activities, from the contexts the user works in, we build the necessary links between documents.

In this paper we discuss how to enhance and contextualize desktop search based on semantic metadata collected from different contexts available and activities performed on a personal computer. We explore three important contexts: electronic mail, folder hierarchies, and web cache. We describe the semantics of these different contexts by appropriate ontologies and show how to extract and represent the corresponding context information as RDF metadata which can be used by a search application together with a full text index of our documents. Based on the above ontologies and metadata, we also show how the web cache context can be further exploited to give a syndicated task specific view upon the activities of the users, by extracting useful data from the web.

To do these, we propose a 3-layer architecture [8] for generating and exploiting the metadata for enhancing desktop resources, as depicted in Figure 1. At the bottom level, we have the physical resources currently available on the desktop. Even though they can all eventually be reduced to files, it is important to differentiate between them based on content and usage context. Thus, we distinguish structured documents, emails, offline web pages, general files and file hierarchies, which miss a lot of contextual information, such as the author of an email or the browsing path followed on a specific web site. We

generate and store these additional facts using RDF metadata, the second conceptual layer of our architecture. Finally, the uppermost layer implements a ranking mechanism over all resources from the previous levels. An importance score is thus computed for each desktop item, supporting ordering of results for the desktop search application.

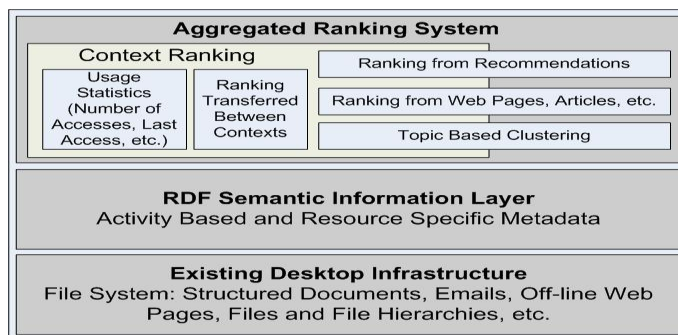


Fig. 1. Semantic Desktop Search Architecture

Existing Desktop Infrastructure. People make use on their desktops of all the available tools: email, file hierarchies and web caches, but sometimes, just using these separate resources is not enough. In order to be able to store and find information we want to have connections between them and use them directly. To keep these relations, we propose to explicitly represent this information in an application independent way as RDF metadata, to enable both enhanced search capabilities, as well as the exploitation of the semantic links between desktop files. (e.g., PDF articles stored from attachments).

RDF Semantic Information Layer. As we just saw, most of the information related to our activities is lost on our current desktops. The aim of this layer is to represent and record this data in RDF annotations associated to each resource. The stored RDF data can be very simple: "author" for an email, "URL" of a web page, or metadata that builds the semantic connections between resources: an email "has attachment" a file, a file can be "stored from" a web page, we can arrive to a visited web page from another.

Aggregated Ranking System. As the amount of desktop items has been increasing significantly, desktop search applications return more and more hits to our queries. If we also take the contextual metadata into account, the results will be even more. A measure of importance which enables us to rank these results is therefore necessary. We propose a basic ranking scheme, based on the ObjectRank [3] idea, enhanced with personalization of trust in resources and links.

The next section presents the main research questions answered in this thesis proposal, illustrated by some related work. The main contributions of our work are further presented, divided into the presentation of the exploited contexts and their ontologies (Section 3), the added ranking system (Section 4) and the current Beagle architecture and how we can achieve a syndicated view upon data using the proposed ontologies, especially from the web cache (Section 5). Finally, Section 6 concludes our work and shows further improvements.

2 Research Questions & Related Work

How Do Users Search on the Desktop? The difficulty of accessing information on our computers has prompted several first releases of desktop search applications, as Google desktop search [16] and the Beagle open source project for Linux [14]. Yet they include *no* metadata whatsoever in their system, but just a regular text-based index. Nor does their competitor MSN Desktop Search [20]. Finally, Apple Inc. has integrated an advanced desktop search application (named *Spotlight Search* [2]) into their new operating system, Mac Tiger. Even though they also intend to add semantics into their tool, only explicit information is used, such as file size, creator or metadata embedded into specific files. In my thesis I focus on how to enhance search using a variety of contextual information often resulting or inferable from explicit user actions or additional background knowledge.

How Do We Rank Resources on the Desktop? Ranking resources on the desktop is difficult as we don't have enough relations between them, as on the web. There have been several papers on ranking in the context of semantic web as Swoogle [9], a search and retrieval system for finding semantic documents on the web. The ranking scheme used in Swoogle uses weights for the different types of relations between Semantic Web documents to model their probability to be explored. However, this mainly serves for ranking between ontologies or instances of ontologies. The importance of semantically capturing user's interests is analyzed in [1]. The purpose of their research is to develop a ranking technique for the large number of possible semantic associations between the entities of interest for a specific query. They define an ontology for describing the user interest and use this information to compute weights for the links among the semantic entities. An interesting technique for ranking the results for a query on the semantic web takes into consideration the inference processes that led to each result [21]. In this approach, the relevance of the returned results for a query is computed based upon the specificity of the relations used when extracting information from the knowledge base. In my thesis I am focusing on PageRank-like algorithm models applicable for the relationships between resources through contextual information.

How Do We Generate the Metadata Structured in Ontologies? Our approach builds upon the idea of a semantic desktop, realizing an unified view upon the resources involved in a certain activity. In [18], the authors describe an approach for personalized content syndication, featuring a central content syndicator instance which answers user requests. Our approach differs from this approach as we do not focus on content brokerage but on metadata brokerage, and incrementally construct metadata which we use for modeling a user's context and preferences. A related approach creating metadata descriptions on behalf of a web extraction process is described in [5]. The author creates RDF descriptions about publication information from dedicated sites in an automated process, as well as new views on the data based on these descriptions and additional background knowledge available for this application.

In the following sections we will discuss the aspects concerning the exploited contexts in the desktop search and the generated ontologies, the ranking of resources and the metadata generation, based on what we described in [12, 8, 13, 11, 7].

3 Contexts & Metadata

In the first two layers of our proposed architecture (see Figure 1), we present the missed contexts available on our desktops, represented by the usage of the physical resources, and the ontologies that emerge from each of these exploitable contexts. In the next sections we differentiate among these resources based on content and usage and present their contextual information in the shape of ontologies.

3.1 Current Desktop Infrastructure and Its Limitations

Today the files on our desktops represent a large amount of data that can no longer be ordered with manual operations such as defining explicit file and directory names. Automatic solutions are needed, preferably taking into account the activity contexts under which each resource was stored / used. In our prototype we focus on the three main working contexts mentioned above, and an additional extended context related to research and scientific publications. All the resources associated to these contexts are currently encountered and valuable information is lost during their utilization.

Web Cache Context. Even though Web search engines are providing surprisingly good results, they still need to be improved to take user context and user actions into account. People sometimes need to search only among the already visited pages, since they remember the subject they have sought, but not so many details regarding the results. We want to annotate each cached web page with additional information both for its basic properties (URL, access date, etc.), as well as more complex ones such as the used in-going and out-going links to other neighboring pages, reflecting the user's surfing behavior. This way, when browsing a certain cached page, enhanced desktop search can also provide information about the context in which that document has been useful for the user, i.e. how it was reached or which links were followed from there.

Publications Context. Research activities represent one of the occupations where the need for contextual metadata is very high. The most illustrative example is the publication itself: Where did this file come from? Did we download it from CiteSeer or did somebody send it to us by email? Which other papers did we download or discuss via email at that time, and how good are they (based on a ranking measure or on their number of citations)? We might remember the general topic of a paper and the person with whom we discussed about it, but not its title. These questions arise rather often in a research environment and have to be answered by an appropriate search infrastructure.

There are also the email and files and file hierarchy contexts that have been explored by us, but not so important for the present work. Due to space limitations, we refer the reader to [7, 8], where we proposed several solutions to enrich the information associated to each resource type presented above. We further present solutions to represent this information and exploit it for desktop search applications.

3.2 RDF Semantic Information Layer

People organize their lives according to preferences often based on their activities. Consequently, desktop resources are also organized according to performed activities and personal profiles. Since, as described above, most of the information related to these

activities is lost on our current desktops, the goal of this layer is to record and represent this data in RDF annotations associated to each resource. Figure 2 depicts an overview image of the ontology that defines appropriate annotation metadata for our contexts.

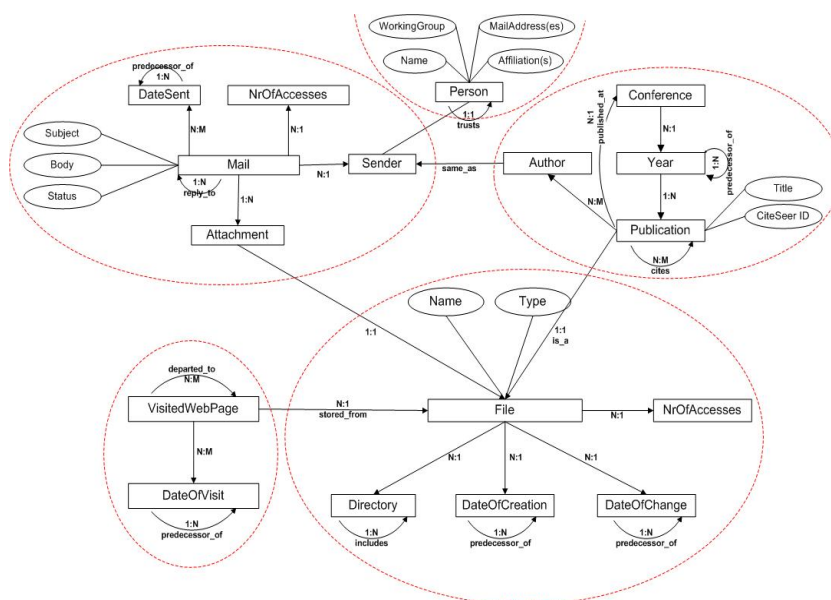


Fig. 2. Contextual Ontology for the Semantic Desktop

For each type of resources we add the basic properties as date of an email or the name of a file, and the more complex ones: an email has an attachment or thread information related to a "reply to" email. But we also need to build the connections between the resources on the desktop. A visited web page or an attachment of an email can be stored as a file, a publication can cite another. We refer the reader to our previous work [7] describing the ontologies associated to these activity contexts.

4 Ranking Resources

As we have already stated, we need a ranking algorithm to put order among the numerous results we may receive. The following paragraphs describe such a ranking mechanism, based on the PageRank algorithm.

Basic Ranking. Given the fact that rank computation on the desktop would not be possible without the contextual information, which recreates the links among resources, annotation ontologies should describe all aspects and relationships among resources influencing the ranking. The identity of the authors for example influences our opinion of documents, and thus "author" should be represented explicitly as a class in our publication ontology. Besides the important concepts, the user activities have also to be taken into account, which translates into assigning different weights to different contexts.

applying the random surfer model and including all nodes in the base set. The random jump to an arbitrary resource from the data graph is modeled by the vector e , which contains an entry for each resource appearing in the data graph. In the original PageRank/ObjectRank formula, the probability of reaching a certain resource through a random jump is evenly distributed among the resources, and therefore, the e vector has only 1 values. Parameter d in the equation represents the dampening factor and is usually considered 0.85. According to the formula, a random surfer follows one of the outgoing links of the current page with the probability d , and with probability $(1-d)$, he jumps to a random page from the web graph. The r vector stores the ranks of all resources in the data graph. These rankings are computed iteratively until a certain threshold is reached.

A is the adjacency matrix which connects all available instances of the existing context ontology on one's desktop. The weights of the links between the instances correspond to the weights specified in the authority transfer annotation ontology. Thus, when instantiating the authority transfer annotation ontology for the resources existing on the users desktop, the corresponding matrix A will have elements which can be either 0, if there is no edge between the corresponding entities in the data graph, or they have the value of the weight assigned to the edge determined by these entities, in the authority transfer annotation ontology, divided by the number of outgoing links of the same type. Additionally, in the case of publications, for the rank computation we also take into account the ratings extracted from the CiteSeer database, with the aid of our publication metadata generator. These values are used as seed values for the calculation of the personalized rankings.

5 Metadata Generation

In this section we present the modality to generate metadata structured in ontologies, using our Beagle prototype. We also show how to syndicate these metadata into semantic views according to the user's activities and interests.

5.1 Current Beagle Architecture

Our current prototype is being built on top of the open source Beagle desktop search infrastructure, which we extended with additional modules: metadata generators, which handle the creation of contextual information around the resources on the desktop, and a ranking module, which computes the ratings of resources so that search results are shown in the order of their importance. The advantage of our system over existing desktop search applications consists in both the ability of identifying resources based on an extended set of attributes – more results, and of presenting the results according to their ranking – to enable the user to quickly locate the most relevant resource.

The main characteristic of our Beagle⁺⁺ ("++" being used to denote our extensions) architecture is metadata generation and indexing on-the-fly, triggered by modification events generated upon occurrence of file system changes. Events are generated whenever a new file is copied to hard disk or stored by the web browser, when a file is deleted or modified, when a new email is read, etc. Much of this basic notification functionality is provided on Linux by an inotify-enabled Linux kernel, which is used by Beagle.

5.2 Extending Beagle with Metadata Generators

Depending on the type and context of the file / event, metadata generation is performed by appropriate metadata generators, as described in Figure 4. These applications build upon an appropriate RDFS ontology as shown in [7], describing the RDF metadata to be used for that specific context. Generated metadata are either extracted directly (e.g. email sender, subject, body) or are generated using the appropriate association rules plus possibly some additional background knowledge. All of these metadata are exported in RDF format, and added to a metadata index, which is used by the search application together with the usual full-text index.

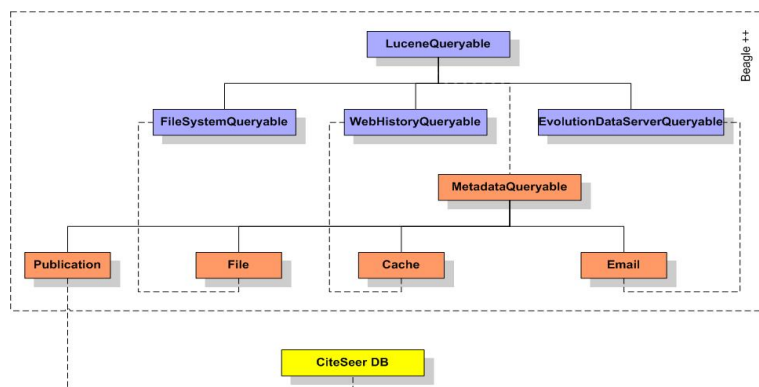


Fig. 4. Beagle Extensions for Metadata Support

The architecture of our prototype environment includes four prototype metadata generators according to the types of contexts described in the previous sections. We added new subclasses of the `LuceneQueryable` class dealing with the generation of metadata for the appropriate contexts (Files, Web Cache, Emails and Publications). The annotations we create include the corresponding elements depicted in the ontology graph in Figure 2. They are described in detail in [7, 8].

5.3 Task Specific Semantic Views: Extracting and Integrating Contextual Metadata from the Web

People structure their (work) lives according to their main activities and, emerging from these daily activities, browsing history is an important mirror of their information seeking behavior. Typically, when people search for information on the web, they do not rely on only one source of information, but many. For example, a user will look on CiteSeer for the papers that are cited by a specific paper as well as the ones citing it, and on DBLP for the conference that the paper was published at and search for more papers in the same track. So, in general, what people try to do is to collect useful information from many sites and manually syndicate this information on their desktop, hoping to have a better view over all relevant information available for specific tasks.

This useful information in the desktop search context, as we have discussed in [7], is available in browser caches. However, from these pages stored as HTML documents, it is very difficult to extract the relevant information automatically. What we really need is to have this information represented in a structured form, automatically transformed into the relevant task specific RDF context metadata, as specified by the task specific ontologies. We propose to automatically extract relevant context information from web sites and automatically syndicate this context information into context metadata specified by task specific ontologies, a global view over available context information.

Relevant Data and Transformation Steps We can partition the process behind our model into distinct steps, further detailed in the rest of this paper. The first step takes care of the extraction of information from various web sites, each web site having a specific schema for their content, as discussed in [11]. The data retrieved in an XML format will contain the relevant web page information for each context and will be transformed into RDF data using XSLT. After this transformation, the data is syndicated and materialized (based on appropriate mapping rules) into task specific semantic views, and can be used to answer queries based over these views.

Schemas for Web Page Content. Data driven HTML pages contain two kinds of information: the repetitive structure of the page (data items listed as rows or structured in distinct sections), and what is presented within this structure (the actual information). The first type reflects the structure of the database that was used to generate the web page. All pages generated from databases and a lot of other ones repeat the same structural items so that we can recognize different information items rather easily: in the case of CiteSeer, the information about scientific publications is often presented in the same style and includes title of the paper, year of publication, authors and cited papers.

As a first step, we need local schemas for the web pages interesting for a specific context. Appropriate collections of web pages share some structure for presenting the content, e.g. all pages from CiteSeer about publications belong to a class, together with the web pages that are associated to this page by dedicated links (the according “cited by” web pages). In a (manual) preparation step, we analyze each of these collections for expressive metadata, and design a small, local ontology which describes the objects of discourse of each of these collections. We can then harvest information about entities and their different attributes. For example, for papers we have publications and conferences and their attributes. In all these cases, the information on the web page is semi-structured, allowing us to construct metadata from these web pages in a semi-automated manner, based on the reconstructed schema and an appropriate query on the HTML page which extracts information according to that schema.

Task Specific Semantic Views. Depending on the tasks and context the user is working in, his context includes all relevant information from the local schemas. This context can be represented by a task specific semantic view integrating the relevant data from the local sources we discussed in the previous section. This semantic view specifies all metadata needed of this context, and is described using ontologies. Let us take a look at such an ontology for CiteSeer, as described in Figure 2. If we compare this ontology and the local CiteSeer ontology in [11], we see that some attributes are omitted from the more web page-specific ontologies, such as “Co-citations” retrieved

from the CiteSeer web page, and that some attributes have different names, even though they represent the same information. So this ontology represents a specific view on the local data sources, appropriate for the task context.

Transforming Web Page Content Information into Task Specific Metadata.

Combining data from different sources and providing the user a unified view of these data is known as the “data integration” problem [23, 19, 6]. The set of sources (e.g. the visited web pages) contain the relevant data, while a global schema (e.g. the task specific ontologies) provides a unified view of the underlying sources. For modeling the relation between the sources and the global schema we will use the global-as-view approach [10, 15], which describes the mappings between local sources and the global schema as a set of assertions $g \rightsquigarrow q_S$, where g represents an element of the global schema and q_S a query over the sources. Such mappings explicitly specify how to query the local sources for each element contained in a query over the global schema, or alternatively, how to materialize the global schema based on the instances from the local data sources.

Using these mappings, we can materialize instances of the task specific ontologies when the user browses new web pages, or reformulate queries over the task specific ontologies during search time. Obviously, the second alternative is not really useful in our context as users have come to expect nearly instantaneous access to search results from web search engines. The global database is thus constructed by merging the important information from the relevant sources of information, i.e. the information extracted from the web pages browsed are merged into the global database containing our activity driven metadata as specified by the task specific ontologies. We can easily see that the data is not only a projection or a subset of the data provided by one site, but another representation of the information. When we map from the global to the local level, we can have different transformations from the different local schemas.

Metadata Extraction and Transformation Now that we know how our global ontologies look like and how the data extracted from the web pages is structured, we just need to describe how exactly we transform data from local data sources into RDF instances corresponding to the global ontologies. We need the following two steps to go from web pages to contextual metadata: *extract* task related metadata from distributed, inhomogeneous sources into local schemas and then *transform* this gathered metadata into one, common context schema.

Extraction of Web Information Using Lixto. We use the Lixto Toolkit [17] for handling the extraction of the structured information contained in web pages. The Visual Wrapper from Lixto [4] provides methodology and tool for the visual and interactive generation of query wrappers - programs, that automatically extract data from semi-structured data sources like web pages and transform them into XML. The extractor, using as input an HTML document and a previously constructed program, generates as its output a pattern instance base, a data structure which encodes the extracted instances as hierarchically ordered trees and strings. In a second step, the extracted XML data is then transformed to RDF using an XSLT script.

Transformation into Task Specific Semantic Views Based on the Mapping Rules. After the extraction and transformation of data according to our local schemas, we then have to transform these data into the global schema, which gives us a unified

view on *all* the local sources that we can query for each scenario. The ontologies described in Section 3.2 provide these task specific semantic views, specifying the final format for the contextual metadata for each context.

The translation between the local properties and relations identified on the web sites (e.g., CiteSeer local ontology) and the properties and relations that are specified in the syndicated ontology is facilitated by the mapping rules. They are necessary for all items we want to keep for the global view. In order to materialize our task specific semantic views, we translated the mapping rules into TRIPLE rules [22].

Triggering These Transformation Steps. The queryable responsible for the web cache annotation is WebHistoryQueryable (see Figure 4). Each URL typed in the web browser that is not in the cache will be transmitted by Beagle⁺⁺ to the Lixto wrapper that harvests the data according to the appropriate local ontologies. The XML data are then accordingly translated and stored into a RDF file, and indexed appropriately. If the URL is in the cache, the relevant metadata will be displayed.

6 Conclusions and Further Work

We presented two main contributions that enhance traditional desktop search, focusing on how regular text-based desktop search can be enhanced with semantics / contextual information and ranking. Searching for resources will not only retrieve explicit results but also items inferred from the users' existing network of resources and contextual information. Maintaining the provenance of information can help the search engine take into account the recommendations from other users and thus provide more retrieved results, as we suggested in [13, 12]. The ranking module, by exploiting contextual information, improves retrieval and presentation of search results, providing more functionality to desktop search.

We also discussed how relevant data can be automatically extracted from web sites visited by the user during his work and syndicated into task specific semantic views, the contextual information relevant for specific tasks and contexts. This contextual information can be exploited to enhance desktop search beyond full-text indexing, leading to more search results as well as to richer result representation. Additionally, we intend to investigate in more detail how to incrementally update views whenever the content of revisited web pages has changed, in order to keep our contextual information consistent.

There are quite a few additional interesting contexts that are worth investigating: metadata embedded in multimedia files, the relations between objects embedded within each other (a presentation including pictures, tables, charts, etc.), or chat history. A further interesting question we want to investigate in the future is how to learn contextual authority transfer weights from user feedback on ranked search results.

References

1. B. Aleman-Meza, C. Halaschek, I. Budak Arpinar, and A. Sheth. Context-aware semantic association ranking. In *Semantic Web and Databases Workshop Proceedings*, 2003.
2. Apple spotlight search. <http://developer.apple.com/macosx/tiger/spotlight.html>.

3. A. Balmin, V. Hristidis, and Y. Papakonstantinou. Objectrank: Authority-based keyword search in databases. In *VLDB*, Toronto, September 2004.
4. R. Baumgartner, S. Flesca, and G. Gottlob. Declarative information extraction, web crawling, and recursive wrapping with lixto. In *6th International Conference on Logic Programming and Nonmonotonic Reasoning*, Vienna, Austria, 2001.
5. R. Baumgartner, N. Henze, and M. Herzog. The Personal Publication Reader: Illustrating Web Data Extraction, Personalization and Reasoning for the Semantic Web. In *European Semantic Web Conference ESWC 2005*, Heraklion, Greece, May 29 - June 1 2005.
6. A. Cali, D. Calvanese, G. De Giacomo, and M. Lenzerini. On the expressive power of data integration systems. In *21st Int. Conf. on Conceptual Modeling*, 2002.
7. P.-A. Chirita, R. Gavriloaie, S. Ghita, W. Nejdl, and R. Paiu. Activity based metadata for semantic desktop search. In *In Proceedings of the 2nd European Semantic Web Conference*, Heraklion, Greece, May 2005.
8. P.-A. Chirita, S. Ghita, W. Nejdl, and R. Paiu. Semantically enhanced searching and ranking on the desktop. In *Submitted for publication, L3S Technical Report*, 2005.
9. L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. C. Doshi, and J. Sachs. Swoogle: A search and metadata engine for the semantic web. In *Proceedings of the 13th ACM Conference on Information and Knowledge Management*, Washington, DC, November 2004.
10. H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. D. Ullman, V. Vassalos, and J. Widom. The TSIMMIS approach to mediation: Data models and languages. *Journal of Intelligent Information Systems*, 8(2):117–132, 1997.
11. S. Ghita, N. Henze, and W. Nejdl. Task specific semantic views: Extracting and integrating contextual metadata from the web. In *Submitted for publication, L3S Technical Report*, 2005.
12. S. Ghita, W. Nejdl, and R. Paiu. Semantically rich recommendations in social networks for sharing and exchanging semantic context. In *Proceedings of the Ontologies in P2P Communities Workshop, ESWC*, Heraklion, Greece, May 2005.
13. S. Ghita, W. Nejdl, and R. Paiu. Semantically rich recommendations in social networks for sharing, exchanging and ranking semantic context. In *Proceedings of ISWC*, Galway, Ireland, November 2005.
14. Gnome beagle desktop search. <http://www.gnome.org/projects/beagle/>.
15. C. Hian Goh, S. Bressan, S. Madnick, and M. Siegel. Context interchange: new features and formalisms for the intelligent integration of information. *ACM Transactions on Information Systems*, 17(3):270–270, 1999.
16. Google desktop search application. <http://desktop.google.com/>.
17. G. Gottlob, C. Koch, R. Baumgartner, M. Herzog, and S. Flesca. The Lixto Data Extraction Project — Back and Forth between Theorie and Practice. In *ACM Symposium on Principles of Database Systems (PODS)*, volume 23. ACM, June 2004.
18. W. Kießling, W.-T. Balke, and M. Wagner. Personalized content syndication in a preference world. In *EnCKompass Workshop on E-Content Management*, Eindhoven, Netherland, 2001.
19. A. Y. Levy, A. O. Mendelzon, Y. Sagiv, and D. Srivastava. Answering queries using views. In *Proceedings of the 14th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 95–104, San Jose, Calif., 1995.
20. Msn desktop search application. <http://beta.toolbar.msn.com/>.
21. N. Stojanovic, R. Studer, and L. Stojanovic. An approach for the ranking of query results in the semantic web. In *ISWC*, 2003.
22. Triple, an rdf rule language. <http://triple.semanticweb.org/>.
23. J. D. Ullman. Information integration using logical views. *Theoretical Computer Science*, 239(2):189–210, 2000.