Speaking Words of WISDOM: Web Intelligent Search based on DOMain ontologies*

Sonia Bergamaschi[†], Paolo Bouquet[‡], Paolo Ciaccia[§], and Paolo Merialdo[♯]

[†]Università degli Studi di Modena e Reggio Emilia
[‡]Università degli Studi di Trento
[§]Università degli Studi di Bologna
[♯]Università degli Studi Roma Tre

Abstract. in this paper we present the architecture of a system for searching and querying information sources available on the web which was developed as part of a project called WISDOM. key feature of our proposal is a distributed architecture based on (i) the peer-to-peer paradigm and (ii) the adoption of domain ontologies. at the lower level, we support a *strong*, ontology-based integration of the information content of a bunch of source peers, which form a so-called *semantic peer*. at the upper level, we provide a *loose*, mapping-based integration of a set of semantic peers. we then show how queries can be efficiently managed and distributed in such a two-layer scenario.

1 introduction

the WISDOM (Web Intelligent Search based on DOMain ontologies)[1] project aims at studying, developing and experimenting methods and techniques for searching and querying information sources available on the Web. The main goal of the project is the definition of a software framework that allows computer applications to leverage the huge amount of information contents offered by Web sources (typically, as Web sites). In the project we assume that the number of sources of interest might be extremely large, and that sources are independent and autonomous one each other. These factors raise significant issues, in particular because such an information space implies heterogeneities at different levels of abstraction (format, logical, semantics). Providing effective and efficient methods for answering queries in such a scenario is the challenging task of the project.

The cardinal idea of the project is to develop a framework that supports a flexible, and yet efficient integration of the semantic content. Key feature of our proposal is a distributed architecture based on (i) the peer-to-peer paradigm and (ii) the adoption of domain ontologies. By means of these ingredients we separate the integration of information sources in two levels: at the lower level we perform and manage a *strong* integration, which involves the information content of a bunch of sources to form a *semantic peer*; an ontology describes the (integrated) information offer of a semantic

* This research has been partially funded by the italian MIUR PRIN WISDOM project (2004-2006).

[1] `http://www.dbgroup.unimo.it/wisdom`

peer. At the upper level, we provide a *loose* integration among the information offered by a set of semantic peers; namely we build a network of peers by means of semantic mappings among the ontologies of a set of semantic peer. When a query is posed against one given semantic peer, it is suitably propagated towards other peers among the network of mappings.

Paper Outline The paper is organized as follows. Section 2 presents an overview of the WISDOM architecture. Section 3 describes how information offered by HTML sources can be wrapped and included in a peer. Section 4 illustrates how we address the issue processing queries.

2 The WISDOM architecture: overview

Current peer-to-peer (P2P) networks support only limited meta-data sets such as simple filenames. Recently a new class of P2P networks, so called schema based P2P networks have emerged (see [1, 26, 10, 30]), combining approaches from P2P as well as from the data integration and semantic web research areas. Such networks build upon peers that use metadata (ontologies) to describe their contents and semantic mappings among concepts of different peers' ontologies. In particular, in Peer Data Management Systems (PDMS) [26] each node can be a data source, a mediator system, or both; a mediator node performs the semantic integration of a set of information sources to derive a global schema of the acquired information.

As stated in a recent survey [3], the topic of semantic grouping and organization of content and information within P2P networks has attracted considerable research attention lately (see, for example, [12, 14]). In super-peer networks [38], metadata for a small group of peers is centralized onto a single super-peer; a super-peer is a node that acts as a centralized server to a subset of clients. Clients submit queries to their super-peer and receive results from it; moreover, super-peers are also connected to each other, routing messages over this overlay network, and submitting and answering queries on behalf of their clients and themselves. The *semantic overlay clustering* approach, based on partially-centralized (super-peer) networks [29] aims at creating logical layers above the physical network topology, by matching semantic information provided by peers to clusters of nodes based on super-peers.

In [7] an approach which combines the schema-based and super-peer network approaches, that is a *schema-based super-peer* network (called SEWASIE network from the UE IST project where it was developed - www.sewasie.org) has been proposed. It is organized into a two-level architecture: the low level, called the peer level (which contains a mediator node), the second one, called super-peer level, (which integrates mediators peers with similar content).

The WISDOM architecture follows a schema based super-peer network architecture. Figure 1 shows the main architectural elements of a WISDOM semantic peer. A WISDOM semantic-peer P_i is composed by a set of heterogeneous information sources $S_{i1}, S_{i2}, ..., S_{in}$ available on the Web. Typically an information source corresponds to the set of pages published by a Web. A WISDOM semantic peer integrates these sources with a traditional wrapper-mediator architecture, as follows.

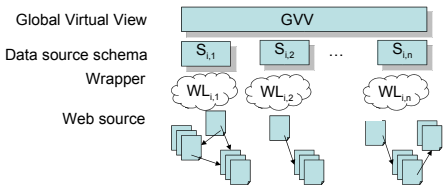

Fig. 1. WISDOM Semantic peer

2.1 Wrappers

Every information source S_{ij} is associated with a wrapper W_{ij}, whose goal is to make the data access method transparent to the upper layers. A wrapper offers a logical schema S_{ij} against which the upper layers can pose queries. For a single page, a wrapper consists of a program that extract data from the HTML code, and organizes results in a more structured format, e.g. an XML document. For a Web site, a wrapper offers a structured and uniform view of the data published in the site: besides data extraction primitives, the wrapper encodes the a description of the hypertext paths to reach data of interest. We will face the issue of inferring wrappers for whole web sites by extending the methods presented in [21].

2.2 Semantic Peer Global Virtual View

The set of schemas offered by wrappers associated with the sources participating one semantic peer are conciliated and integrated in a Global Virtual View (GVV). The goal of the GVV is to provide an integrated, coherent and consistent view of the information contents offered by the sources of the semantic peer.

The GVV is created by the MOMIS system framework, a semi-automatic tool which creates the GVV as a domain ontology based on a description logics layer [9, 6]; the GVV global classes are *annotated* with respect to a lexical ontology (Wordnet [32]).

Intra-peer mappings specify how the GVV relates to the local sources managed by the semantic peer. We follow a GAV (Global as a View) strategy: each element of the GVV is described as a view q_N over the source schemas. In the WISDOM project the integration designer can implicitly define q_N by using: the *Full Disjunction* operator (that has been recognized as providing a natural semantics for data merging queries); and extending it with *Data Conversion Functions* from local to global attributes, *Join Conditions* among local classes and *Resolution Functions* for global attributes (to solve data conflicts of local mapped attribute values).

We follow and extend the approaches proposed in [34, 25] for computing Full Disjunction and in [33] for resolution functions.

2.3 Semantic Peer Overlay Network

We build an overlay network of semantic peers in order to allow our framework to retrieve information of interest even outside the semantic peer that received the query.

Figure 2 illustrates the main architectural elements that frame such a network. The overall idea is to associate with every semantic peer an ontology Ont_i, which describes the information offered by the semantic peer itself. A network of semantic peers is thus build by defining mappings between the ontologies of a set of semantic peers (we consider binary mappings).

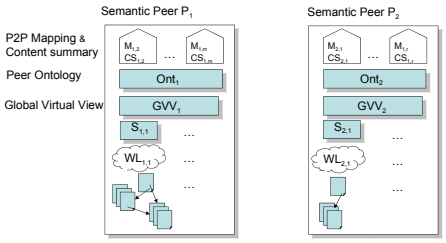

Fig. 2. WISDOM semantic peer network

The ontology of a semantic peer provides an intensional description of its information contents. The ontology also represents the interface of the semantic peer: relationship with other semantic peer are built through the ontology. Also, we foresee that users interact (i.e. query and browse information) with the system by means of the ontologies of peers.

The main idea is to define the Semantic Peer Ontology starting from the GVV; the GVV of a set of local sources is built in a semi-automatic way with the MOMIS approach [9, 6]; as a consequence, generally it is a *flat schema*, semantically annotated, referred as *Bootstrap Ontology*; the Semantic Peer Ontology is obtained by eventually enriching this Bootstrap Ontology by:

1. modifying its concepts in order to define them at a higher abstraction level
2. defining new concepts, such as attributes, classes, relationships.
3. aligning and integrating it with other ontologies, such as Classification Schemata and pre-existing ontologies

2.4 Peer-to-Peer Mapping and Query Processing

To frame a network of semantic peers we adopt *peer-to-peer mappings*: a semantic peer-to-peer mapping, denoted $M_{i,j}$, is a relationship between the ontology Ont_i of the semantic peer P_i, and the ontology Ont_j of the semantic peer P_j. Intuitively a mapping $M_{i,j}$ allows the rewriting of a query posed against the ontology Ont_i into a query over the ontology Ont_j. Mappings will be computed by extending the methodology for p2p semantic coordination presented in [11]. The main idea is that discovering mappings across ontologies requires a combination of lexical knowledge, world knowledge and structural information (how concepts are arranged in a specific ontology). This information is used in a methodology called *semantic elicitation*, which builds a formal representation of the information represented by a concept in an ontology (or even more

frequently in "light weight ontologies", like for example taxonomies of classification schemas). This formal representation is then used to infer relations with formal objects in other ontologies by automated reasoning systems (e.g. SAT solvers, description logic reasoners, and so on); the choice of the system depends on the expressiveness of the representation which is built during the elicitation phase. Notice that, since the discovery of a mapping may depend on knowledge which is available only locally at some peer, it may be the case that computed mappings are asymmetric, i.e. a semantic peer P_i may discover a mapping from its local ontology Ont_i to an external ontology Ont_j, whereas the corresponding semantic peer P_j may fail to compute (or even explicitly reject) the existence of a corresponding mapping from its point of view.

By means of peer-to-peer mappings, a query received by a given peer can be ideally extended to every peer for which a mapping is defined. However, it is not always convenient to propagate a query to *any* peer for which a mapping exists. For example, it can be inefficient to include in the query processing peers having a limited extension of the concepts involved by the query. To overcome this issue, we associate every peer-to-peer mapping with a *content summary*. Given a pair of semantic peers for which it exists a peer-to-peer mapping, the content summary associated with such a mapping provides quantitative information about the extension of the concepts in the source ontology that can be found through the mapping in the target semantic peer. A simple example of the information provided by a content summary is the cardinality, in the target peer, of the concepts of the source ontology. To further speed-up the execution of queries, the WISDOM query processor will also implement execution strategies able to quickly compute the "best" answers to a query.

3 Wrapping Large Web Sites

A large number of Web sites contain highly structured regions. These sites represent rich and up-to-date information sources, which could be used to populate WISDOM semantic peers. However, since they mainly deliver data through intricate hypertext collections of HTML documents, it is not easy to access and compute over their data.

To overcome this issue, several researchers have recently developed techniques to automatically infer web wrappers [4, 15, 19, 37], i.e., programs that extract data from HTML pages, and transform them into a machine processable format, typically in XML. The developed techniques are based on the observation that many web sites contain large collections of structurally similar pages: taking as input a small set of sample pages exhibiting a common template, it is now possible to generate as output a wrapper to extract data from any page sharing the same structure as the input samples.

These proposals represent an important step towards the automatic extraction of data from web data sources. However, as argued in [4, 19], intriguing issues arise when scaling up from the single collection of pages to whole sites. The main problems, which significantly affect the scalability of the wrapper approach, are how to identify the structured regions of the target site, and how to collect the sample pages to feed the wrapper generation process. Presently, these tasks are done manually.

To overcome this limitation we are studying techniques to addresses these issues, making it feasible to automatically extract data from large data intensive web sites. The

goal is to develop a system that infers a model that describes at the intensional level the overall structure of the target site in terms of classes of pages and navigational paths among them. We expect that structurally similar pages are grouped into classes, and that the link between pages can be grouped into classes of links between classes. Also, we aim at generating the model efficiently, i.e. by visiting a small portion of the target site: while building the model, the system should choose the pages to visit in order to infer a complete model for the site.

consider Figure 3: given a large web site composed by thousands of interconnected page (left), we aim at producing a model, that describes at the intensional level the structure of the site (right).

To give an intuition of the idea, consider the official FIFA 2002 world cup web site,[2] whose roughly 60,000 pages contain information about teams, players, matches, and news. These pages are strongly interconnected, as depicted in Figure 3 (left). However both the contents and the topology of the site are structured in a regular way: there is one page for each player, one page for each team, and so on; the links between pages reflect the semantic relationships between contents: for instance, every team page contains links to the pages of its players. The model we aim to infer describes these features at the intensional level, as depicted in Figure 3 (right).

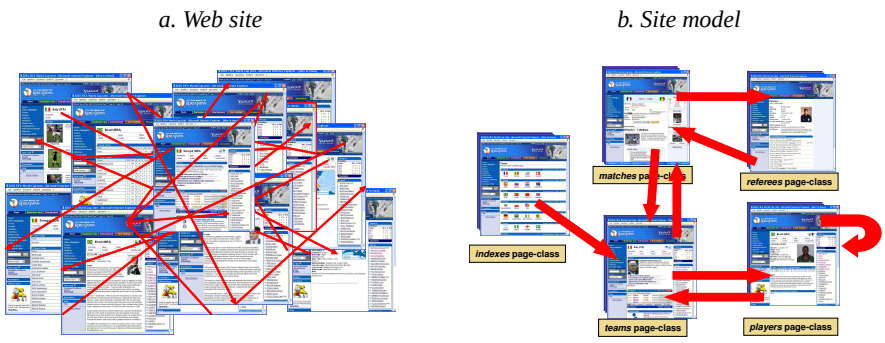

Fig. 3. Intensional description of a web site

Based on such a site model we can infer a library of wrappers, ideally one wrapper for each class of pages, with the help an external wrapper generator.[3] The model, together with the associated wrappers, can then be used to continuously extract data from the target web site.

A key observation in our approach is that in data-intensive web sites links reflect both the internal structure of pages and the topological structure of the site. Whenever two (or more) pages contain links that share the same layout and presentation properties, then it is likely that the two pages share the same structure. This can be more easily

[2] `http://fifaworldcup.yahoo.com/02`

[3] We use ROADRUNNER, our wrapper generator module [19, 20, 18].

observed by grouping the links into (possibly singleton) *collections*: each collection is characterized by uniform layout and presentation properties. The layout and presentation properties of each link collection can be described, for example, in terms of their paths on the DOM tree. Consider for example Figure 4, which shows the web page of a player again from the FIFA web site: every player page (they are all generated from the same template) contains the same link collections shown in this page (e.g. one link collection on the right leading to the other player of the same team, and one link collection on the top pointing to other pages about its team, and so on).

Fig. 4. A player page from `fifaworldcup.com`

In addition, we observe that usually links within the same collection are organized either as a list of link pointing to similar pages, or as a tuple of links leading to pages with different structures. Again from our example, we have that every player page contains one link collection on the right, which is a list of links to other player pages, one link collection on the top pointing to several different kinds of page about a team ("overview", "profile", "stats" ...), etc..

Our approach for inferring a model describing the structure of a web site builds on the above observations. We model pages as objects containing tuples and list of links; a *page class* models a set of pages; links are typed: the type corresponds to the class of the target page. To infer a model for a web site, we consider its pages, and group them into classes considering several alternative partitions, which are then ranked according to a metrics of quality. The inference process is performed incrementally. Starting from a given entry point (e.g. the home page), which becomes the first member of the first class in the model, the model is refined by exploring its boundaries to gather new pages. At each iteration, a link collection from the model outbound is selected and its target pages are explored. In order to reduce the number of pages to visit, after each download we make a guess on the class of remaining pages. If looking at the pages already downloaded there is sufficient evidence that the guess is right, the remaining

pages of the collections are assigned to classes without actually fetching them. The process iterates until all the link collections are typed with a known class.

4 Query Processing

Processing a query in WISDOM consists of several steps, as described in the following.

4.1 Query formulation

To ease the user in the task of formulating queries, a graphical user interface is provided that allows queries to be specified with respect to the ontology of the peer the user is connected to ("target ontology"). Besides specifying conditions (= hard constraints) that objects have to satisfy, a user query might also include *preferences* [16, 28, 35], which are used by the WISDOM query processor to determine the relevance of objects and, consequently, to return only the most relevant results to the user. As an example, if a user is interested in budget accommodations in highly-rated hotels, this translates into a preference specification that aims to both minimizing the price and maximizing the rating of the hotels.

The result of a query Q with a preference specification $pref$ is the set of objects, reachable from the target peer by navigating its mappings, that better comply with $pref$. Since in the general case $pref$ defines a strict partial order on objects, the result of Q consists of all and only the "undominated" objects, that is, objects for which no better alternative is obtainable. The set of such best objects can be conveniently reduced if the user specifies a limit, say k, to the cardinality of the result, in which case only k, non-deterministically chosen, of such best objects will be part of the result.

The output of this first step is a query represented in an internal, graph-based, form.

4.2 Query rewriting and peers selection.

By means of peer-to-peer mappings, a query received by a target peer can in principle be rewritten and propagated to every (source) peer for which a mapping with the target peer is defined. However, for efficiency reasons, this is not always desirable, since the number of possible rewritings is typically too high. In particular, peers having only a few, if none at all, relevant results to return (e.g., no budget accommodations to offer) should not be involved in the query processing task. Although by means of a specific "preference directive" it is always possible to instruct the query processor to consider all rewritings, in WISDOM we deliberately do not insist on *completeness* of results. This is in line with the observation that (quoting from [2]):

> *At Web scale [giving a complete answer to every query] is unfeasible and query execution must move to a probabilistic world of evidence accumulation and away from exact answers.*

To deal with the problem of selecting only relevant peers we characterize every peer-to-peer mapping with a *content summary*, CS, that is, a synopsis of the source

peer contents. In the simplest form a CS includes the cardinalities, in the source peer extension, of the concepts in the target ontology (e.g., how many hotels the source peer knows about). This is recursively extended to include also information on the extensions that can be found navigating the network through the source peer, much alike *routing indices* do in schema-less peer-to-peer networks [22]. In practice, CS's store more detailed information in order to allow for a more precise selection of the most relevant rewritings (and, consequently, of the most relevant source peers), such as histograms of the distribution of properties' values (e.g., distribution of hotel prices).

Additional information that is exploited by the query processor to select the most relevant peers to which forward the query include the accuracy of the rewriting, i.e., how much the rewritten query conforms to the original query specification, and other features of the peers, such as network bandwidth and expected response time, that might affect overall performance. Note that this makes it possible to deal in a uniform way with local and remote rewritings, which simplifies the task of query execution.

The output of this phase is a set of "ranked rewritings" $R_1, \ldots, R_m$ for the original query Q, with rewriting R_1 being reputed the "most promising" one to return relevant results. Note that each R_i might still involve more than one peer, i.e., its results are to be determined by joining those returned by the involved peers.

4.3 Query optimization

For each rewriting R_i, and starting with R_1, the query processor applies a set of rule-based transformations so as to obtain a more efficient execution strategy. For instance, classical selection push-down is performed at this step, so as to reduce the amount of data to be transmitted over the network and to be joined across peers. Also, the original preference specification is split across the participating peers. As an example, consider again the query that looks for highly-rated budget accommodations, and assume that rewriting R_i involves peers P_1 and P_2. However, P_1 knows about hotel prices, but has no information on ratings, whereas the opposite is true for P_2. In this case, an optimized execution strategy would ask P_1 to return cheapest hotels and P_2 highest-rated hotels first.

At the end of this step an internal representation of the query is produced that, for each rewriting, consists of an optimized execution plan tree. The leaves of such tree are the actual queries that the query processor will send to the source peers.

4.4 Query execution

To start with, consider the execution of the "best" rewriting, R_1, and assume that R_1 involves the source peers $P_1, \ldots, P_m$, each of which has to process a part of R_1, denoted by $R_{1,i}$ ($i = 1, \ldots, m$). A naïve execution strategy would be to have P_0 collecting all the results, $Res(R_{1,i})$, and then locally performing other operations needed to complete the plan tree. This strategy will perform poorly, because it has to wait for all peers to complete their work before being able to produce the very first result. Further, a large amount of data will be transmitted across the network.

The basic approach to query execution in WISDOM inspires to works developed for joining ranked inputs, that have been applied with success to relational and multimedia

databases, and information retrieval systems [24, 13, 5, 27, 31]. In all these scenarios, and modulo minor differences, we have a set of data sources, each one ranking objects according to a specific *local* criterion, and we wish to determine the overall best objects, i.e., those objects which are ranked higher with respect to a *global* criterion. It is now well understood that for the answer to be computable without retrieving all the data from the underlying sources it is necessary and sufficient that the global ranking criterion is positively correlated with the local ones (i.e., doing better locally cannot worsen the overall performance of an object). If this condition is met, then execution can exploit the local rankings to halt as soon as it can be proved that nothing better that what seen so far can be obtained.

In a network of peers such as that in WISDOM things get more complex, and above techniques are properly extended to deal with this increased complexity. As an example, we have faced the problem of incomplete information (e.g., how to rank a hotel with no rating?) and provided a semantics that still guarantees an efficient computation of the result. From the algorithmic point of view, our retrieval strategy differs from that presented in [17] for computing ranked full disjunctions since we consider that preferences define a strict partial order on objects (rather than only linear ones as in [17]).

4.5 Query execution in a semantic peer

A query posed against the GVV retrieves data from the integrated sources: according to the GAV strategy, queries are unfolded by taking into account the view q_N: every atom of the global schema is expanded, substituting its description with subqueries expressed with references to the local schemata on the basis of the defined mappings. The results defined from the subqueries onto the local schema is integrated and reconciliated in a global answer on the basis of q_N (see [7, 8]).

4.6 Browsing the results

Visualizing the results of a WISDOM query faces a common problem, that is, to guarantee a satisfactory compromise between expressivity and domain-independence when visualizing and navigating RDF-like graphs. Here expressivity is meant as the capability of delivering an intuitive representation of knowledge and some tailored navigation primitives to end-users working in a given application domain, while domain-independence aims to accomplish a high degree of reusability. Most existing tools, such as KAON [36] and WebOnto [23] , favor domain-independence and represent entities in a way that is closer to the abstract form used to formally define them. This is familiar to knowledge engineers (a narrow category of end-users) but not to domain experts. Indeed, though domain-specific formalisms have a lower degree of reusability, they provide graphically richer constructs allowing for a representation that is closer to how entities appear in the application domain. An approach to address this issue is to build a flexible framework in which reusable components realize the domain-independent tasks in generating a friendly presentation of a piece of knowledge. In WISDOM we have developed M-FIRE, a configurable framework for easily instantiating visualization and navigation systems based on the adoption of custom *metaphors*. Metaphors drive the

process through which visual representations are obtained for RDF documents, and define how queries are generated upon user actions. This allows users to perform semantic browsing by relying on intuitive concept representations and to interact in a simple manner with complex knowledge.

5 Conclusions

We presented the architecture of a system for searching and querying information sources available on the Web which was developed as part of a project called WISDOM. The main feature of the proposed architecture is a distinction in two layers. At a lower level, we imagine that groups of peers, each of which provides some information content, may decide to share one or more domain ontologies, and therefore achieve a strong level of integration of what they can offer outside; this is done through by building Global Virtual Views across local schemas, using a GAV approach. Such a strongly integrated bunch of peers is what we called a semantic peer. At a higher level, we imagine that different semantic peers (i.e. groups of peers which do not share their domain ontologies) can achieve a looser form of integration by computing and using mappings across heterogeneous ontologies. These mappings can be used to rewrite queries based on some ontologies into queries which conform to different ontologies, and therefore to retrieve information which is annotated and organized in heterogeneous ways. We discussed in detail issues related to wrapping large web sites, and to optimizing the distribution of queries by providing content summaries of what is available at a given peer.

References

1. K. Aberer, P. Cudré-Mauroux, and M. Hauswirth. The chatty web: emergent semantics through gossiping. In *WWW*, pages 197–206, 2003.
2. S. Abiteboul, R. Agrawal, P. Bernstein, and et al. The Lowell Database Research Self-Assessment. *Communications of the ACM*, 48(5):111–118, May 2005.
3. S. Androutsellis-Theotokis and D. Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv.*, 36(4):335–371, 2004.
4. A. Arasu and H. Garcia-Molina. Extracting structured data from web pages. In *ACM SIGMOD International Conf. on Management of Data (SIGMOD'2003), San Diego, California*, pages 337–348, 2003.
5. I. Bartolini, P. Ciaccia, V. Oria, and M. T. Özsu. Flexible Integration of Multimedia Subqueries with Qualitative Preferences. *Multimedia Tools and Applications Journal*, 2006. To appear.
6. D. Beneventano, S. Bergamaschi, F. Guerra, and M. Vincini. Synthesizing an integrated ontology. *IEEE Internet Computing*, 7(5):42–51, 2003.
7. D. Beneventano, S. Bergamaschi, F. Guerra, and M. Vincini. Querying a super-peer in a schema-based super-peer network. In *In proceedings of the 3rd VLDB International Workshop on Databases, Information Systems and Peer-to-Peer Computing*, 2005.
8. D. Beneventano and M. Lenzerini. Final release of the system prototype for query management. *Sewasie, Deliverable D.3.5, Final Version, available at http://www.dbgroup.unimo.it/pubs.html*, Apr. 2005.
9. S. Bergamaschi, S. Castano, M. Vincini, and D. Beneventano. Semantic integration of heterogeneous information sources. *Data Knowl. Eng.*, 36(3):215–249, 2001.

10. P. A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu. Data management for peer-to-peer computing : A vision. In *WebDB*, pages 89–94, 2002.
11. P. Bouquet, L. Serafini, and S. Zanobini. Peer-to-peer semantic coordination. *Journal of Web Semantics*, 2(1), 2005.
12. J. e. a. Broekstra. A metadata model for semantics based peer-to-peer systems. In *Proc. of the 1st WWW Int. Workshop on Semantics in Peer-to-Peer and Grid Computing (SemPGRID 2003)*, Budapest, Hungary, May 2003.
13. N. Bruno, L. Gravano, and A. Marian. Evaluating Top-k Queries over Web-Accessible Databases. In *Proceedings of the 18th International Conference on Data Engineering (ICDE 2002)*, pages 369–382, San Jose, CA, USA, Feb. 2002.
14. S. Castano, A. Ferrara, S. Montanelli, E. Pagani, and G. Rossi. Ontology-addressable contents in p2p networks. In *Proc. of the 1st WWW Int. Workshop on Semantics in Peer-to-Peer and Grid Computing (SemPGRID 2003)*, Budapest, Hungary, May 2003. http://www.isi.edu/ stefan/SemPGRID/proceedings/proceedings.pdf.
15. C.-H. Chang and S.-C. Lui. Iepad: Information extraction based on pattern discovery. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5*, pages 681–688, 2001.
16. J. Chomicki. Preference Formulas in Relational Queries. *ACM Transactions on Database Systems (TODS)*, 28(4):427–466, 2003.
17. S. Cohen and Y. Sagiv. An Incremental Algorithm for Computing Ranked Full Disjunctions. In *Proceedings of the 24th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'05)*, Baltimore, MD, USA, June 2005.
18. V. Crescenzi and G. Mecca. Automatic information extraction from large web sites. *Journal of the ACM*, 51(5), September 2004.
19. V. Crescenzi, G. Mecca, and P. Merialdo. ROADRUNNER: Towards automatic data extraction from large Web sites. In *International Conf. on Very Large Data Bases (VLDB 2001), Roma, Italy, September 11-14*, pages 109–118, 2001.
20. V. Crescenzi, G. Mecca, and P. Merialdo. Roadrunner: Automatic data extraction from data-intensive web sites. In *ACM SIGMOD International Conf. on Management of Data (SIGMOD'2002), Madison, Wisconsin*, 2002.
21. V. Crescenzi, P. Merialdo, and P. Missier. Clustering pages based on their structure. *Data & Knowledge Engineering, Elsevier*, 54:279–299, 2005.
22. A. Crespo and H. Garcia-Molina. Routing indices for peer-to-peer systems. In *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS 2002)*, pages 23–32, Vienna, Austria, July 2002.
23. J. Domingue, E. Motta, and O. Garcia. *Knowledge Modelling in WebOnto and OCML: A User Guide*. Knowledge Media Institute, Milton Keynes, UK, 1999.
24. R. Fagin, A. Lotem, and M. Naor. Optimal Aggregation Algorithms for Middleware. In *Proceedings of the 20th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'01)*, pages 216–226, Santa Barbara, CA, USA, May 2001.
25. C. A. Galindo-Legaria. Outerjoins as disjunctions. In R. T. Snodgrass and M. Winslett, editors, *SIGMOD Conference*, pages 348–358. ACM Press, 1994.
26. A. Y. Halevy, Z. G. Ives, J. Madhavan, P. Mork, D. Suciu, and I. Tatarinov. The piazza peer data management system. *IEEE Trans. Knowl. Data Eng.*, 16(7):787–798, 2004.
27. I. F. Ilyas, W. G. Aref, and A. K. Elmagarmid. Joining Ranked Inputs in Practice. In *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB 2002)*, pages 950–961, Hong Kong, China, Aug. 2002.
28. W. Kießling. Foundations of Preferences in Database Systems. In *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB 2002)*, pages 311–322, Hong Kong, China, Aug. 2002.

29. A. Löser, F. Naumann, W. Siberski, W. Nejdl, and U. Thaden. Semantic overlay clusters within super-peer networks. In K. Aberer, V. Kalogeraki, and M. Koubarakis, editors, *DBISP2P*, volume 2944 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2003.
30. A. Löser, W. Siberski, M. Wolpers, and W. Nejdl. Information integration in schema-based peer-to-peer networks. In J. Eder and M. Missikoff, editors, *CAiSE*, volume 2681 of *Lecture Notes in Computer Science*, pages 258–272. Springer, 2003.
31. S. Michel, P. Triantafillou, and G. Weikum. KLEE: A Framework for Distributed Top-k Query Algorithms. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB 2005)*, Trondheim, Norway, Sept. 2005.
32. A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.
33. F. Naumann and M. Häussler. Declarative data merging with conflict resolution. In C. Fisher and B. N. Davidson, editors, *IQ*, pages 212–224. MIT, 2002.
34. A. Rajaraman and J. D. Ullman. Integrating information by outerjoins and full disjunctions. In *PODS*, pages 238–248. ACM Press, 1996.
35. R. Torlone and P. Ciaccia. Which Are My Preferred Items? In *AH2002 Workshop on Recommendation and Personalization in eCommerce (RPeC02)*, pages 1–9, Malaga, Spain, May 2002.
36. R. Volz, D. Oberle, S. Staab, and B. Motik. KAON SERVER - A Semantic Web Management System. In *Proceedings of the Twelfth International World Wide Web Conference (WWW2003)*, Budapest, Hungary, May 2003.
37. J. Wang and F. Lochovsky. Data-rich section extraction from html pages. In *Proceedings of the 3rd International Conference on Web Information Systems Engineering (WISE 2002), 12-14 December, Singapore*, pages 313–322. IEEE Computer Society, 2002.
38. B. Yang and H. Garcia-Molina. Designing a super-peer network. In U. Dayal, K. Ramamritham, and T. M. Vijayaraman, editors, *ICDE*, pages 49–. IEEE Computer Society, 2003.