# Database-Inspired Search

David Konopnicki and Oded Shmueli

IBM Haifa Research Laboratory
Haifa, Israel
davidko@il.ibm.com

Computer Science Department, Technion
Haifa, Israel
oshmu@cs.technion.ac.il

## Abstract

"W3QL: A Query Language for the WWW", published in 1995, presented a language with several distinctive features. Employing existing indexes as access paths, it allowed the selection of documents using conditions on semi-structured documents and maintaining dynamic views of navigational queries. W3QL was capable of automatically filling out forms and navigating through them. Finally, in the SQL tradition, it was a declarative query language, that could be the subject of optimization.

Ten years later, we examine some current trends in the domain of search, namely the emergence of system-level search services and of the semantic web. In this context, we explore whether W3QL's ideas are still relevant to help improve information search and retrieval. We identify two main environments for searching, the enterprise and the web at large. Both environments could benefit from database-inspired integration language, and an execution system that implements it.

## 1 Introduction

In 1995, we published "W3QL: A Query Language for the WWW". The goal of the W3QL language was to automate search and retrieval tasks utilizing the (then) existing web search infrastructure, namely full-text search indexes such as Lycos[1] and Infoseek[2]. W3QL was realized in the context of W3QS, a system that also provided various graphic and programming interfaces.

This paper retrospectively examines W3QL, some of the current trends affecting advanced searching (not all, given the limited scope of the paper), and outlines a view of a likely future. In that future, we argue, there is a place for a modernized version of W3QL, to integrate information from a myriad of sources, including objects, documents, semantic information, XML and other text data.

W3QL had several features that were distinctive at the time. Employing the existing indexes as access paths, it allowed the selection of documents that are inherently semi-structured, specifying conditions on document features such as author and title, and maintaining continuous dynamic views of navigational queries. W3QL was capable of automatically filling out forms and navigating to the underlying resources "hiding" behind forms (the so-called "hidden web"). In that respect, it could be used to specify sophisticated personal crawlers. On the other hand, in the SQL tradition, W3QL was a declarative query language that offered opportunities for optimization.

Ten years later, the search landscape has greatly evolved. Search is ubiquitous and is considered a fundamental feature of any computing platform. From the desktop to the internet, through enterprise intranets, the search "giants" are engaged in a fight for control of the search infrastructure. The goal is to ease, as well as to control, access to the mountains of information available on desktop computers, intranets and networks.

Another important development is the emergence of the semantic web effort. Its underlying idea is to equip web resources with semantic information that can be understood by automated tools. Such semantic information may exist within the enterprise and on the internet in a distributed fashion. It also introduces deduction and distributed ontology extensions.

Search techniques have also influenced relational database technology. Today, most relational database vendors support querying XML (semi-structured) data and provide some form of integration with full-text indexes. Furthermore, in the realm of XML querying,

**Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2005**

[1] www.lycos.com

[2] www.infoseek.com

XQuery is being extended with support for full-text search operators [3].

We distinguish two main environments for searching. One is at the internet at large. While it is possible to cache a large portion of the web, it is impractical to cache it all. This is due to the fact that (a) web information is rapidly changing, (b) the whole web is not necessarily visible to crawlers, (c) some of the data reside in databases and are presented dynamically, and (d) legal and privacy constraints. With the emergence of the semantic web, and with distributed semantic data and servers of various deduction capabilities, this situation is likely to continue. As a consequence, W3QL's basic notion of adding querying capabilities on top of search indexes is still relevant.

The other environment is enterprise-centric. Here, while the web at large may still be an important resource, a large portion of the needed information is (a) in-house, (b) accessible, (c) controllable - in terms of format, content and versioning. This provides ample opportunities for efficient and thorough information extraction.

A database-inspired approach for searching would: (a) utilize a data model, (b) employ a formal, declarative and optimizable query language, (c) enable data to be treated from multiple viewpoints, and (d) utilize optimization techniques and supporting storage structures. In the emerging search landscape, a database-inspired approach in conjunction with the existing and planned tools (based on information retrieval), database techniques and the emerging semantic web technologies, seems very relevant.

# 2 W3QL and W3QS

## 2.1 A Short History

In 1994, NCSA Mosaic was the latest web browser. There were approximately 25,000 web sites and, when clicking on a hypertext link, loading the desired page could take several minutes. In order to ease and automate search and retrieval tasks on the internet, we defined W3QL, a SQL-like query language. We also implemented W3QS, a system for executing W3QL queries. W3QS provided several useful services such as continuously maintained views and various user interfaces for non-programmers at different levels of sophistication (see Figure 1).

The essential underlying idea of W3QL was to look at the WWW from a database perspective:

- W3QL was, most importantly, a database language. First,W3QL employed a directed graph-based data model in which web pages are nodes and links are edges. Second, there was a clear separation between the "what" (is the desired result) and the "how" (to compute it). And so, in the SQL tradition, W3QL was a declarative query language that offers opportunities for optimization. For example, one could start navigation at various potential starting points, each providing a number of URLs, and a query evaluation strategy was needed to pursue the search. In [39], we explored how web navigational queries may be optimized.

- W3QL used existing full-text indexes as search starting points. Whereas potentially the search scope is the whole web, practically, only URLs visible through indexes (or known to users a-priori) could serve as navigation starting points. So, W3QL used search services as database-like indexes. For example, if one wanted to look at Technical Reports in CS departments, to obtain starting points one could use 'Technical Report' and 'CS', and pose a query to search engines.

- Recognizing that it is impossible to devise a meaningful schema for WWW information, W3QL was inherently extendible. Any (user-defined) Unix tool could be invoked in order to evaluate conditions and filter pages. As default, the PERLCOND tool was used for expressing conditions on semi-structured file formats such as (n1 is a variable that corresponds to a page):

```
n1.format eq "Latex File"
   && n1.section[3].content =~ /zoo/
```

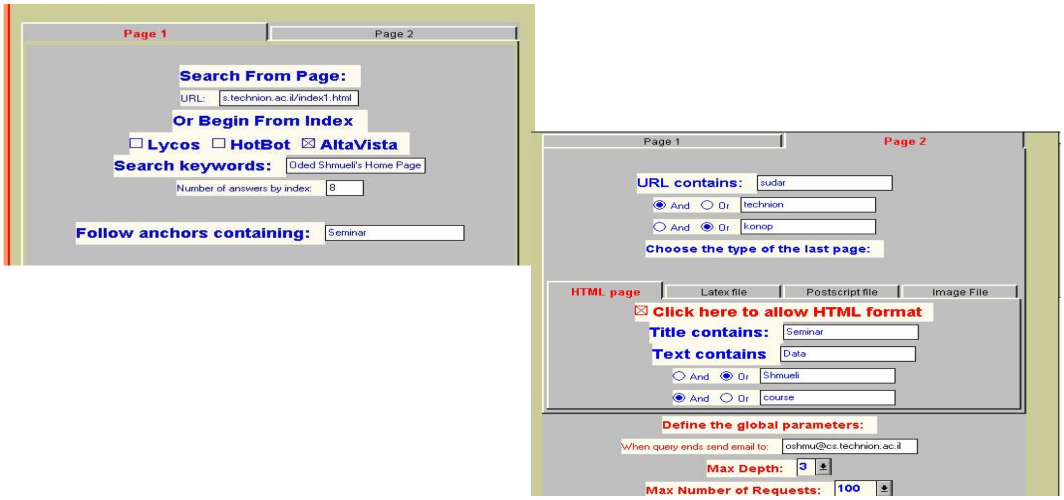
- W3QL was capable of specifying how forms that are encountered during navigation are to be filled. The W3QS system maintained a database of encountered forms and was able to fill out newly encountered forms based on how similar forms had been filled out in the past (form understanding has recently been addressed in [60]).

- W3QL supported a rudimentary level of abstraction of data formats. For example, if a condition on the author attribute of a document was specified in a query, W3QS would, for each file encountered in the search, (a) determine its format, and (b) extract the author attribute as encoded in the format at hand.

Using W3QL, one could automate tedious retrieval tasks such as "go to the West university home page, navigate to every faculty member home page, and retrieve from there all papers in PDF format published in 2005". This query could also be maintained as a materialized view. Thus, W3QL could be considered as a language to specify personalized crawlers.

## 2.2 Some Contemporary Systems

W3QL was one of several search systems that attempted to use database techniques to query the web,



Search From Page:

URL: s.technion.ac.il/index1.html

Or Begin From Index

□ Lycos □ HotBot ☒ AltaVista

Search keywords: Oded Shmueli's Home Page

Number of answers by index: 8

Follow anchors containing: Seminar

URL contains: sudar

● And ○ Or technion

○ And ● Or konop

Choose the type of the last page:

☒ Click here to allow HTML format

Title contains: Seminar

Text contains Data

○ And ● Or Shmueli

● And ○ Or course

Define the global parameters:

When query ends send email to: oshmu@cs.technion.ac.il

Max Depth: 3

Max Number of Requests: 100

Figure 1: A non-programmer user interface for specifying W3QL queries

hypertext and semi-structured data. Predecessor systems included [1], [44], [9] and [19]. Lorel was one of the first languages for querying semi-structured data [2]. Another was UnQL [15] with a somewhat different approach and capabilities (its data model is value-based and it employs structural recursion).

The first generation languages for web search (as classified by [29]) also included WebSQL, which is close in spirit to W3QL. WebSQL used a more structured approach [47]; in particular, it encoded the link structure within a binary relation. It also separated the notions of local site and external links. The second generation languages included logic-based languages such as Weblog [41] and object-oriented and deductive languages such as Florid [45].

The second generation also included languages with web site generating capabilities such as WebOQL [7] and StruQL [27]. In Araneus [46, 8], a database approach was used to generate, or restructure, web sites, using the languages Ulixes (for data extraction from pages into relations) and Penelope (for hypertext generation). A general architecture for querying web content is presented in [20].

For a survey of the state of affairs at the late 1990s, see [29]. Search engines architectures are described in [6, 12].

# 3 Present Trends

## 3.1 Limitations of Search Services

W3QS was a pre-Google tool. Still, at the time there were some fairly advanced search engines such as Lycos, Hotbot and Altavista. Today's search services such as Google[3], Yahoo[4] and MSN Search[5] are more advanced and comprehensive. Some of the changes in capabilities include:

- The way queries are specified has evolved. In addition to some form of boolean conditions on keywords and field conditions (on author, title etc. . . ), one can restrict the search to some internet domain, country, language, file types etc. . .

- Some search services provide search result clustering. For example, in Teoma[6], if one searches for 'bar', the search results are refined into the following categories: Bar Association, State Bar, Bar Codes, Law School, Bar Examiners, Barcode Software, Bar Review, Nightclubs etc. Some provide a combination of topic links and conventional page listings [7]. Interestingly, categorization of query results has recently migrated to relational databases [17].

- Search services provide specialized sub-services to query users' desktops, the web, images, newsgroups, books content etc...[8].

- An organized topic summary (utilizing on-line dictionaries, encyclopedias and other resources), categorization, as well as starting point for further explorations, are provided by Answers.com[9].

Still, the major service provided by search engines is the ranking of results. Even with their impressive quickness and apparent usefulness, search engines today still have deficiencies. Some of these are inherent, due to business structure, privacy or cost constraints.

1. The ranking of results may be swayed towards links to paying customers. As there is no indepen-

[3] www.google.com

[4] www.yahoo.com

[5] search.msn.com

[6] www.teoma.com

[7] Ask.com, clusty.com

[8] www.google.com/intl/en/options

[9] www.answers.com

dent auditing, consumers have no way of knowing how “objective” the ranking is.

2. Research shows that one browses the first page of results and that the other dozens of pages are usually ignored [43]. Therefore, the search must accommodate the “average user” and may be ineffective for more sophisticated purposes.

3. There still seems to be no way to provide search engine feedback about the utility of the results in a secure, privacy-preserving and non-intrusive way.

4. Expanding on this last point, search engines are not aware of the context in which the search is carried out. That is, what precisely is the knowledge the user is looking for? A search engine can only estimate the user’s intentions based on the search terms used and assuming “an average user”. Presumably, had it known the search context or search workflow, it could have provided more useful and focused information.

   A step in the direction of understanding the search context is the new “Yahoo Mindset” experimental search service[10]. This service allows users to influence result ranking based on whether the desired search results are more commercial or more informational.

5. Search engines mostly return “raw” data as they appear on the indexed pages. This requires users to parse each individual result page, get used to its formatting, organization and language, and obtain the needed information.

   A step in the direction of returning more “standardized”, object-like results is the “Google Local” service[11]. Using this service, when one searches for “pizzeria”, rather than obtaining web pages containing this keyword, one obtains pizzeria objects having fields such as address and telephone, as well as a link to a web page for further information. This approach in essence combines Google’s search index with a structured Yellow Pages-like service.

The W3QL concept of providing a user-tailored service that utilizes the existing search services is still relevant. Such an approach enables users to:

- Define more complex search and retrieval tasks (that go beyond first page browsing).

- Combine results from various search services (similar to what meta-search engines do[12]).

- Automatically process many more results than could conceivably be manually browsed.

- Explore search neighborhoods rather than just the supplied result pages.

- Combine proprietary techniques during the search, e.g., for analyzing natural language.

- Format results as needed for the task at hand.

Efforts in this direction are already taking place. The Aquaint project[13] attempts to address these shortcomings, aiming to support in-depth analysis tasks within the Intelligence community. It aims to provide question understanding and answering, against heterogeneous collections of structured and unstructured information of multiple media, within a context. It also addresses the organization and presentation of results.

### 3.2 The Consolidation of Search Services

Search is now ubiquitous. The explosion in the amount of available information necessitates powerful search tools. The average PC is equipped with 80GB of disk space. A gigantic amount of information is available on the internet and in enterprise intranets. Providing employees with the information they need is a constant challenge for IT departments. Therefore, search is viewed more and more as a *system service.* That is, instead of having different applications (the file system, email etc. . . ) managing their own search index, search services maintain integrated indexes on behalf of all applications. These system-level search services are, and are expected to be, available as stand-alone search appliances on the network (e.g., Verity’s Ultraseek[14], IBM’s Omnifind[15] or Google Search Appliance[16]); within operating systems and file systems (such as the Microsoft Windows indexing service and as envisioned in [30]); in enterprise portal platforms (for example, the WebSphere Portal Search Engine); and even in network storage systems.

One can view these search services as *integration services.* On the desktop, these services integrate data from all applications. In the enterprise intranet, the search services integrate information from the various collaboration tools (blogs, wikis, email servers, document management systems etc. . . ) and from enterprise systems. Thus, we call this new generation of search indexes, Search and Integration indexes (SIIs). SIIs are mainly based on information retrieval technologies, and therefore do not require a mediated schema.

[10] mindset.research.yahoo.com

[11] www.google.com/lochp

[12] e.g., www.mamma.com

[13] www.ic-arda.org/InfoExploit/aquaint

[14] www.verity.com/products/ultraseek

[15] www-306.ibm.com/software/data/-integration/db2ii/editions_womnifind.html

[16] www.google.com/enterprise/gsa

To be effective, SIIs provide APIs that allow applications to upload their data into the search indexes (e.g., the Google desktop SDK[17]) and provide crawlers for extracting data from a wide variety of data sources: email servers, directory servers, web content management systems and relational databases.

The strategic importance of these ubiquitous system-level search services cannot be overemphasized. In years to come, SIIs are likely to be the essential means of access to the structured, semi-structured and unstructured mountains of data produced in personal computers, company intranets and on the internet.

### 3.3 Merging Document and Object Retrieval

Another important trend is the evolution of search and retrieval systems from being document-centric to object-centric. As mentioned, several internet and intranet search tools return objects rather than web pages. In intranets, such tools return information pertaining to documents, people, locations, computers, software packages etc... extracted from the various enterprise systems.

This trend is exemplified by the new IBM Unstructured Information Management Architecture (UIMA) [13]. In this architecture, documents pass through a customizable analysis pipeline. In this pipeline, analysis engines are used to annotate documents. Annotations can range from simple, for example "*this* text fragment is the name of a person", to complex, for example "*this* text fragment is a shop whose owner is the person whose name is in *that* text fragment ...".

When a document is annotated, its annotations form a set of semi-structured objects. At the end of the analysis pipeline, a full-text index is constructed in which documents are indexed together with their annotations (expressed in XML). So, this mixture of unstructured and semi-structured information can be queried [16].

Annotation systems may benefit from information extraction (IE) techniques. These techniques are attractive within (structurally) well-understood domains, e.g., the extraction of bibliographic references from scientific papers for a site such as Citeseer[18]. The more sophisticated IE systems rely on machine learning methodologies whereby the system is constructed using training data and is then used to extract information from unseen documents. Machine learning techniques used for IE include Hidden Markov Models (HMMs), Support Vector Machines (SVMs) [33], Conditional Random Fields (CRF) [50] and more. Recent experience with some IE tools is reported in [35].

### 3.4 The Semantic Web

The semantic web is a vision for a future web in which information is categorized and made comprehensible by automated tools of various kinds. This vision is in the first phase of a lengthy realization process. Along the way, various components are being introduced which provide additional capabilities.

The foundation is the resource description framework (RDF[19]) which provides a basic level of semantic tagging. RDF's basic structure is a graph composed of triples of the form $(subject, predicate, object)$. Such a triple declares a relationship between the subject and the object. The vocabulary that can be used in building an RDF graph is defined in RDF Schema[20]. Once RDF graphs are defined, they may be queried using several query languages such as RDQL[21] and SPARQL[22]. An examination of the relationships between RDF graphs and their expressiveness appears in [32].

The next piece in the semantic web puzzle is OWL, an ontology definition language[23]:

1. OWL can be used to provide semantics that can be understood by automated tools. The underlying data may then be queried, e.g., using a language such as OWL-QL [28].

2. OWL allows reasoning to be applied based on defined ontologies, which facilitate the deduction of new facts based on existing ones.

3. OWL is designed to be extendible in a distributed fashion.

The semantic web is starting to materialize. Tools for manipulating and managing semantic data are becoming available. For example, Jena 2[24] is a Java framework for writing semantic web applications. Jena 2 includes: an RDF API, ARP (a RDF/XML parser), persistence, a reasoning subsystem, an ontology subsystem, and a RDQL implementation.

Currently, the most popular uses of RDF are RSS and FOAF. RSS [42] is used for syndicating news and blogs. Utilizing RSS, web sites can publish metadata describing their latest changes. Users can employ a search engine for RSS feeds, such as Feedster[25], to retrieve the latest published news on a particular subject. The RSS mechanism is particularly useful for fast-changing sites, such as blogs and news sites, which search indexes cannot crawl often enough to be kept up-to-date.

[17] desktop.google.com/developer.html

[18] citeseer.ist.psu.edu

[19] www.w3.org/RDF

[20] www.w3.org/TR/rdf-schema

[21] www.w3.org/Submission/RDQL

[22] www.w3.org/TR/rdf-sparql-query

[23] www.w3.org/TR/owl-features

[24] www.hpl.hp.com/semweb/jena.htm

[25] www.feedster.com

FOAF (Friend-of-a-friend) is an OWL-described ontology for "creating and using machine-readable homepages that describe people, the links between them and the things they create and do"[26]. Here's an example of a FOAF file describing Oded Shmueli produced via foaf-a-matic[27]:

```
...
</foaf:PersonalProfileDocument>
<foaf:Person rdf:nodeID="me">
<foaf:name>Oded Shmueli</foaf:name>
<foaf:title>Prof.</foaf:title>
<foaf:givenname>Oded</foaf:givenname>
<foaf:family_name>Shmueli</foaf:family_name>
<foaf:mbox_sha1sum>...</foaf:mbox_sha1sum>
<foaf:homepage
     rdf:resource="www.cs.technion.ac.il/~oshmu"/>
<foaf:phone rdf:resource="tel:+972-4-829-4280"/>
<foaf:schoolHomepage
    rdf:resource="..."/>
<foaf:knows>
    <foaf:Person>
        <foaf:name>David Konopnicki</foaf:name>
        <foaf:mbox_sha1sum>...</foaf:mbox_sha1sum>
    </foaf:Person>
```

FOAF highlights two important aspects:

- First, simple, semantically unambiguous data formats allow the building of useful global information systems, such as community networks, in a completely distributed fashion.

- Second, meta-data need to be constructed. To make the semantic web a reality, simple and friendly authoring tools such as foaf-a-matic are essential, together with tools for the automatic production of metadata such as UIMA.

Advanced web search engines that make use of RDF semantic information and ontologies have recently appeared, e.g. Swoogle [21].

Semantic information and knowledge-bases are expected to be distributed over the web. It is unlikely that it will be possible to cache these knowledge-bases in the way ordinary web pages are cached by search indexes today. Deduction may be unpredictable in terms of the number of results to be expected, the time to produce them and their level of binding. In addition, similar results may be produced from various sources. Protocols, such as those provided by OWL-QL [28] provide handles for exercising some control over the query evaluation process. However, a tool for information extraction will need a component for managing and optimizing the interactions that are enabled by the OWL-QL protocol, as well as interactions with other sources.

## 4 An Outlook: The Search Integration Challenge

The search landscape is being transformed by the emergence of search and integration indexes (SIIs), merging data from multiple applications and systems. These services will not only provide raw text. They will also offer a mixture of objects, semi-structured data, semantic descriptions and text. Most of these components will be created automatically by analysis engines (such as UIMA's), and some will be RDF objects (or documents) and enable advanced reasoning.

There will be several types of SIIs. SIIs will operate in desktop computers, in servers of various divisions of an enterprise, in the storage arrays managed by Internet Service Providers and in several public internet services such as Google. Next, we examine how SIIs will be queried and how they will be managed.

### 4.1 Querying

The paradigm of retrieving objects based on textual relevance is indeed powerful. However, in the newly emerging environment more is needed. There is a need for a high-level integration language that possesses capabilities for processing both structured and semi-structured data in addition to information retrieval techniques.

This integration language, a modern version of W3QL (and W3QS), may be based on a number of ideas, algorithms and technologies:

1. Flexible querying capabilities. An important characteristic of semi-structured data is the lack of full schema information. This motivates the inclusion of path expression constructs in many languages for semi-structured data. Flexible querying has been extensively investigated in recent years, and various semantics have been explored. For example, the ideas of semiflexible and flexible matching were introduced in [37]. Semantics for partial answers, the OR-semantics and the weak semantics were considered in [36]. Flexible XML querying, employing query pattern relaxation as well as full-text search conditions, is described in [4]. Advanced search and extraction languages and systems need such a similar precise and flexible way of handling information at the integration language level.

2. Ranking composition. Fagin and Wimmers [26], in the context of the Garlic project [18], developed a method for transforming a formula for combining scores based on values $x_1, \ldots, x_n$ into a formula for combining these scores when the $x_i$s are associated with weights $w_i$s. Intuitively, the $w_i$ weight reflects the importance of the $i'th$ score. The formula transformation has some desirable properties [26]. In the context of web searching

[26] www.foaf-project.org

[27] www.ldodds.com/foaf/foaf-a-matic

it has been used to ascribe importance to pages based on weights that users associate with search terms [22]. This transformation technique may be used by an advanced search tool to dynamically decide on the most profitable navigation direction.

3. Top-$k$ queries. Consider a set of multidimensional objects where each dimension is associated with an attribute (e.g., color, shape). Objects are scored, per dimension, according to the similarity of their attribute (field) value in that dimension, to a query specified value for that attribute (e.g., 'red', 'round'). A monotonic score aggregation function produces, for each object, an overall score based on the scores of the various attributes.

   Suppose we are given, for each attribute, an index list of objects, in decreasing order of score for that attribute. We wish to obtain $k$ objects with the best overall score. Fagin's algorithm ($FA$) addressed this problem [23]. It later evolved into Fagin's threshold algorithm ($TA$) [23, 24, 25], discovered independently by [31, 49]. These algorithms solve this problem by deciding, as early as possible while scanning the index lists, that the $k$ top scoring objects have been determined. $FA$ and $TA$ employ random access to obtain scores for objects. Variations have been considered that restrict access to sorted accesses only, or in which only some lists may be accessed in sorted order. The deterministic guarantees of $TA$ have been extended to probabilistic ones, see for example [54].

   Top-$k$ querying capabilities appear to be an essential component in searching in an environment that includes many search indexes, that may score objects on various attributes or combinations of attributes, and in which various authority scores may be assigned to the various sources.

4. Ranked query results. Processing and optimizing relational database queries whose results are ranked is described in [34]. Ranking on the semantic web has recently been considered in [5]. Semantic associations are ranked, where a semantic association is essentially a sequence of associations. An important feature of [5] is modulating the importance of a semantic association based on the context of the search, which ranges from 0 (conventional) to 1 (discovery).

5. $k$ Nearest Neighbors (KNN). Essentially, this includes a set of capabilities for locating the $k$ closest neighbors to a query point in a multidimensional space. The "curse of dimensionality" makes this a difficult problem [55]. Many techniques and data structures have been proposed to solve this problem over the years, for example [11, 58]. The problem formulation has been extended in various directions, e.g. [53]. An index structure to support both similarity range search and KNN search has recently been described in [59]. This set of capabilities is important in providing approximate solutions, in locating similar objects, and in supporting object fusion.

6. Similarity ranking. Two additional capabilities are required to be able to combine information (i.e., essentially computing joins) on entities managed by different applications that may be represented differently in different SIIs.

   - One of the most important requirements of such a language is to integrate a notion of object similarity [38, 10].
   - The ability to discover and rank semantic associations between entities [5].

7. Preferences. To be effective, a querying tool needs to make decisions that involve various parameters. A tool should enable a user to express constraints, preferences and tradeoffs pertaining to characteristics of the required information (datedness, accuracy, authority, coverage), as well as operational costs (such as time, space and perhaps even direct monetary ones). The search process may involve tradeoffs among these parameters. Humans, when browsing, make many decisions. An automated tool also needs a decision-making component. Techniques used in electronic commerce based on goal programming ($GP$) [51] that involve specifying 'deals' as goal programs, may prove useful in this regard [52].

The importance of the semantic web technologies in the context of searching is threefold:

- Semantically unambiguous data allow the retrieval of more meaningful results.
- For some domains, there may be a global ontology (essentially equivalent to a global database schema). Data conforming to this global ontology can be distributed at various sites. The combination of information from the various sites may enable the deduction of information that is not obtainable from each information source taken in isolation (e.g., FOAF for community networks). The KSL wine agent [28] demonstrates how a distributed knowledge-base, built according to the semantic web standards, can be used for matching wines to dinner courses.
- Taking the previous idea further, reasoning capabilities allow discovery of facts that are not apparent in the information sources, but may be *deducible* from their combination.

[28] www.ksl.stanford.edu/people/dlm/webont/wineAgent

One interesting domain of application of these technologies is the dynamic construction of workflows to perform a desired task. Given an appropriate ontology, different online services could specify the service they provide, together with the prerequisites they require and the characteristics of their produced results. Thus, given a formal description of a task that must be carried out, it may be possible to deduce a workflow, i.e. sequence of steps, to accomplish the task. Such a technology could be used to dynamically combine services provided in an enterprise or to provide explicit workflow information found in the open web (i.e., *what steps should one take prior to traveling to a particular foreign country, e.g., obtain a visa, get vaccinated etc...*). The work in [40] presents a step in this direction.

Global queries posed against a collection of SIIs need to be optimized. To do so, queries may be translated into a collection of queries, each being posed against particular SIIs. This implies the need for a standardized search and indexing API that gives more control over the computation of results than the APIs available today (e.g., Google Web APIs[29]). Currently, such APIs usually only allow submission of a query and retrieval of the results.

### 4.2 System Management

The management of SIIs poses new challenges in addition to the challenge of querying this blend of raw, semi-structured and structured data. An SII Management System (SI$^2$MS) will perform the types of activities that were extensively explored in the domain of relational databases. Examples are data warehousing (integrating information from a variety of systems) and query processing (deciding how indexes should be combined). Thus, these activities should be re-evaluated in this new context.

On the other hand, SI$^2$MSs introduce new needs, and therefore present new research problems. Among them are the optimization of crawling sessions [57], fast indexes update [14] and full-text index partionning [12], to name a few.

### 4.3 The Emerging Picture

Figure 2 depicts the new search environment:

- System-level indexes used for data search and integration (SIIs) aggregate data from various applications and servers from both the enterprise and the internet.

- Data aggregation is done through analysis pipelines in which data are augmented with semi-structured metadata.

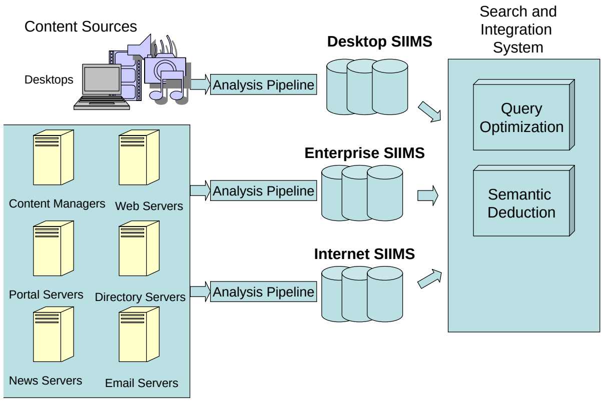


Figure 2: System View

- Several SI$^2$MSs are available at different levels of granularity: in desktop computers, in different parts of the enterprise and, globally, in the internet. These different SIIs may have different query capabilities. Some may be based on information retrieval technology and others may provide semantic web deduction services.

- The search and integration system does not necessarily have its own storage capabilities. It uses a query language a la W3QL to execute queries that combine information from the different SIIs. It utilizes a wide array of ideas, algorithms and technologies from a variety of research domains.

## 5 Conclusion

We offer a plausible future in which search indexes, currently used mostly for textual documents, are used to aggregate data from all kinds of applications and servers. These search and integration indexes (SIIs) contain not only text data, but also a mix of textual, semi-structured and structured data. Some of these data rely on semantic web ontologies. Some are obtained by analyzing the source data within analysis pipelines (such as UIMA). In this context, we have identified several issues related to the management of these SI$^2$MSs (SII Management Systems) and sketched the requirements for a query language used to harness their data.

**Acknowledgment:** We would like to thank Ran El-Yaniv and Yoelle Maarek for their help.

## References

[1] S. Abiteboul, S. Cluet, and T. Milo. Querying and updating the file. In *Proc. VLDB*, pages 73–84, 1993.

[29] http://www.google.com/apis

[2] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. Wiener. The lorel query language for semistructured data. *Journal on Digital Libraries*, 1(1):68–88, 1996.

[3] S. Amer-Yahia, C. Botev, and J. Shanmugasundaram. Texquery: a full-text search extension to xquery. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 583–594, New York, NY, USA, 2004. ACM Press.

[4] Sihem Amer-Yahia, Laks V. S. Lakshmanan, and Shashank Pandit. Flexpath: Flexible structure and full-text querying for xml. In Weikum et al. [56], pages 83–94.

[5] Kemafor Anyanwu, Angela Maduko, and Amit Sheth. Semrank: ranking complex relationship search results on the semantic web. In Allan Ellis and Tatsuya Hagino, editors, *WWW*, pages 117–127. ACM, 2005.

[6] Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, and Sriram Raghavan. Searching the web. *ACM Trans. Inter. Tech.*, 1(1):2–43, 2001.

[7] O. Arocena and A. O. Mendelzon. Weboql: Restructuring documents, databases and webs. In *Proc. ICDE*, pages 24–33, 1998.

[8] P. Atzeni, G. Mecca, and P. Merialdo. To weave the web. In *Proc. VLDB*, pages 206–215, 1997.

[9] C. Beeri and Y. Kornatzky. A logical query language for hypertext systems. In *Proceeding of the European Conference on Hypertext*, pages 67–80, 1990.

[10] Catriel Beeri, Yaron Kanza, Eliyahu Safra, and Yehoshua Sagiv. Object fusion in geographic information systems. In Nascimento et al. [48], pages 816–827.

[11] Stefan Berchtold, Christian Böhm, H. V. Jagadish, Hans-Peter Kriegel, and Jörg Sander. Independent quantization: An index compression technique for high-dimensional data spaces. In *ICDE*, pages 577–588, 2000.

[12] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *WWW7: Proceedings of the seventh international conference on World Wide Web 7*, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[13] A. Z. Broder and A. C. Ciccolo. Towards the next generation of enterprise search technology. *IBM Syst. J.*, 43(3):451–454, 2004.

[14] E.W. Brown, J.P. Callan, and W.B. Croft. Fast incremental indexing for full-text information retrieval. In *Proceedings of the 20th International Conference on Very Large Databases (VLDB)*, pages 192 – 202, Santiago, Chille, September 1994.

[15] Peter Buneman, Mary Fernandez, and Dan Suciu. Unql: a query language and algebra for semistructured data based on structural recursion. *The VLDB Journal*, 9(1):76–110, 2000.

[16] David Carmel, Yoelle S. Maarek, Matan Mandelbrod, Yosi Mass, and Aya Soffer. Searching xml documents via xml fragments. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 151–158, New York, NY, USA, 2003. ACM Press.

[17] Kaushik Chakrabarti, Surajit Chaudhuri, and Seung won Hwang. Automatic categorization of query results. In Weikum et al. [56], pages 755–766.

[18] W. F. Cody, L. M. Haas, W. Niblack, M. Arya, M. J. Carey, R. Fagin, M. Flickner, D. Lee, D. Petkovic, P. M. Schwarz, J. Thomas, M. Tork Roth, J. H. Williams, and E. L. Wimmers. Querying multimedia data from multiple repositories by content: the garlic project. In *Proceedings of the third IFIP WG2.6 working conference on Visual database systems 3 (VDB-3)*, pages 17–35, London, UK, UK, 1995. Chapman & Hall, Ltd.

[19] M. P. Consens and A. O. Mendelzon. Expressing structural hypertext queries in graphlog. In *Proc. Hypertext*, pages 269–292, 1989.

[20] Hasan Davulcu, Juliana Freire, Michael Kifer, and I. V. Ramakrishnan. A layered architecture for querying dynamic web content. In *SIGMOD '99: Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 491–502, New York, NY, USA, 1999. ACM Press.

[21] Li Ding, Tim Finin, Anupam Joshi, Rong Pan, R. Scott Cost, Yun Peng, Pavan Reddivari, Vishal Doshi, and Joel Sachs. Swoogle: a search and metadata engine for the semantic web. In *CIKM '04: Proceedings of the Thirteenth ACM conference on Information and knowledge management*, pages 652–659, New York, NY, USA, 2004. ACM Press.

[22] R. Fagin and Y. Maarek. Allowing users to weight search terms. In *Proc. RIAO Computer-Assisted Information Retrieval*, pages 682–700, 2000.

[23] Ronald Fagin. Combining fuzzy information from multiple systems. *J. Comput. Syst. Sci.*, 58(1):83–99, 1999.

[24] Ronald Fagin. Combining fuzzy information: an overview. *ACM SIGMOD Record*, 31(2):109–118, 2002.

[25] Ronald Fagin, Amnon Lotem, and Moni Naor. Optimal aggregation algorithms for middleware. *J. Comput. Syst. Sci.*, 66(4):614–656, 2003.

[26] Ronald Fagin and Edward L. Wimmers. A formula for incorporating weights into scoring rules. *Theor. Comput. Sci.*, 239(2):309–338, 2000.

[27] M. Fernandez, D. Florescu, J. Kang, A. Levy, and D Suciu. Catching the boat with strudel: Experiences with a web-site management system. In *Proc. SIGMOD*, pages 414–425, 1998.

[28] R. Fikes, P. Hayes, and I. Horrocks. Owl-ql: A language for deductive query answering on the semantic web. Technical Report KSL Technical Report 03-14, Stanford University, 2003.

[29] Daniela Florescu, Alon Levy, and Alberto Mendelzon. Database techniques for the world-wide web: a survey. *SIGMOD Rec.*, 27(3):59–74, 1998.

[30] D. Gifford, P. Jouvelot, M. Sheldon, J. James, and W. O'Toole. Semantic file systems. In *Proceedings of the Thirteenth ACM Symposium on Operating Systems Principles*, pages 16–25, 1991.

[31] Ulrich Güntzer, Wolf-Tilo Balke, and Werner Kießling. Optimizing multi-feature queries for image databases. In Amr El Abbadi, Michael L. Brodie, Sharma Chakravarthy, Umeshwar Dayal, Nabil Kamel, Gunter Schlageter, and Kyu-Young Whang, editors, *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt*, pages 419–428. Morgan Kaufmann, 2000.

[32] Claudio Gutiérrez, Carlos A. Hurtado, and Alberto O. Mendelzon. Foundations of semantic web databases. In Alin Deutsch, editor, *PODS*, pages 95–106. ACM, 2004.

[33] Hui Han, C. Lee Giles, Eren Manavoglu, Hongyuan Zha, Zhenyue Zhang, and Edward A. Fox. Automatic document metadata extraction using support vector machines. In *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, pages 37–48, Washington, DC, USA, 2003. IEEE Computer Society.

[34] Ihab F. Ilyas, Rahul Shah, Walid G. Aref, Jeffrey Scott Vitter, and Ahmed K. Elmagarmid. Rank-aware query optimization. In Weikum et al. [56], pages 203–214.

[35] N. Ireson, F. Ciravegna, M.-E. Califf, A. Lavelli, D. Freitag, and N. Kushmerick. Evaluating machine learning for information extraction. In *Proc. International Conference on Machine Learning (ICML)*, 2005.

[36] Yaron Kanza, Werner Nutt, and Yehoshua Sagiv. Querying incomplete information in semistructured data. *Journal of Computer and System Sciences*, 64(3):655–693, 2002.

[37] Yaron Kanza and Yehoshua Sagiv. Flexible queries over semistructured data. In *PODS '01: Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 40–51, New York, NY, USA, 2001. ACM Press.

[38] Vipul Kashyap and Amit Sheth. Semantic and schematic similarities between database objects: a context-based approach. *The VLDB Journal*, 5(4):276–304, 1996.

[39] D. Konopnicki and O. Shmueli. Www exploration queries. In *Proc. NGITS (LNCS 1649)*, pages 20–39, 1999.

[40] David Konopnicki, Lior Leiba, Oded Shmueli, and Yehoshua Sagiv. A formal yet practical approach to electronic commerce. *Int. J. Cooperative Inf. Syst.*, 11(1-2):93–117, 2002.

[41] L. Lakshmanan, F. Sadri, and I. N. Subramania. A declarative language for querying and restructuring the web. In *Sixth International Workshop on Research Issues in Data Engineering - Interoperability of Nontraditional Database Systems*, 1996.

[42] Juhnyoung Lee and Richard Goodwin. The semantic webscape: a view of the semantic web. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1154–1155, New York, NY, USA, 2005. ACM Press.

[43] Ronny Lempel and Shlomo Moran. Predictive caching and prefetching of query results in search engines. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 19–28, New York, NY, USA, 2003. ACM Press.

[44] A. Y. Levy, D. Srivastava, and K. Kirk. Data model and query evaluation in global information systems. *Journal of Intelligent Information Systems*, 5(2), 1995.

[45] B. Ludäscher, R. Himmeröder, G. Lausen, W. May, and C. Schlepphorst. Managing semistructurd data with florid: A deductive object oriented perspective. *Information Systems*, 23(8):589–613, 1998.

[46] G. Mecca, P. Atzeni, A. Masci, P. Merialdo, and G. Sindoni. The araneus web-base management system. In *Proc. SIGMOD*, pages 544–546, 1998.

[47] G. A. Mihaila, A. O. Mendelzon, and T. Milo. Querying the world-wide web. In *Proc. PDIS96*, pages 80–91, 1996.

[48] Mario A. Nascimento, M. Tamer Özsu, Donald Kossmann, Renée J. Miller, José A. Blakeley, and K. Bernhard Schiefer, editors. *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*. Morgan Kaufmann, 2004.

[49] Surya Nepal and M. V. Ramakrishna. Query processing issues in image (multimedia) databases. In *ICDE*, pages 22–29. IEEE Computer Society, 1999.

[50] Fuchun Peng and Andrew McCallum. Accurate information extraction from research papers using conditional random fields. In *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, 2004.

[51] C. Romero. *Handbook of Critical Issues in Goal Programming*. Pergamon Press, 1991.

[52] O. Shmueli, B. Golany, R. Sayegh, H. Shachnai, M. Perry, N. Gradovitch, and B. Yehezkel. Negotiation platform. *International Patent Application, WO 02077759*, 2001.

[53] Yufei Tao, Dimitris Papadias, and Xiang Lian. Reverse knn search in arbitrary dimensionality. In Nascimento et al. [48], pages 744–755.

[54] Martin Theobald, Gerhard Weikum, and Ralf Schenkel. Top-k query evaluation with probabilistic guarantees. In Nascimento et al. [48], pages 648–659.

[55] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In Ashish Gupta, Oded Shmueli, and Jennifer Widom, editors, *VLDB*, pages 194–205. Morgan Kaufmann, 1998.

[56] Gerhard Weikum, Arnd Christian König, and Stefan Deßloch, editors. *Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, France, June 13-18, 2004*. ACM, 2004.

[57] J. L. Wolf, M. S. Squillante, P. S. Yu, J. Sethuraman, and L. Ozsen. Optimal crawling strategies for web search engines. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 136–147, New York, NY, USA, 2002. ACM Press.

[58] Cui Yu, Beng Chin Ooi, Kian-Lee Tan, and H. V. Jagadish. Indexing the distance: An efficient method to knn processing. In Peter M. G. Apers, Paolo Atzeni, Stefano Ceri, Stefano Paraboschi, Kotagiri Ramamohanarao, and Richard T. Snodgrass, editors, *VLDB*, pages 421–430. Morgan Kaufmann, 2001.

[59] Lei Zhang, Yong Yu, Jian Zhou, ChenXi Lin, and Yin Yang. An enhanced model for searching in semantic portals. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 453–462, New York, NY, USA, 2005. ACM Press.

[60] Zhen Zhang, Bin He, and Kevin Chen-Chuan Chang. Understanding web query interfaces: Best-effort parsing with hidden syntax. In Weikum et al. [56], pages 107–118.