Distributed Indexing/Searching Workshop

Agenda, Attendee List, and Position Papers

Held May 28-19, 1996 in Cambridge, Massachusetts

Sponsored by the World Wide Web Consortium

Workshop co-chairs:
Michael Schwartz, @Home Network
Mic Bowman, Transarc Corp.

This workshop brings together a cross-section of people concerned with distributed indexing and searching, to explore areas of common concern where standards might be defined. The Call For Participation[1] suggested particular focus on repository interfaces that support efficient and powerful distributed indexing and searching.

There was a great deal of interest in this workshop. Because of our desire to limit attendance to a workable size group while maximizing breadth of attendee backgrounds, we limited attendance to one person per position paper, and furthermore we limited attendance to one person per institution.

In some cases, attendees submitted multiple position papers, with the intention of discussing each of their projects or ideas that were relevant to the workshop. We had not anticipated this interpretation of the Call For Participation; our intention was to use the position papers to select participants, not to provide a forum for enumerating ideas and projects. As a compromise, we decided to choose among the submitted papers and allow multiple position papers per person and per institution, but to restrict attendance as noted above. Hence, in the paper list below there are some cases where one author or institution has multiple position papers.

---

[1] http://www.w3.org/pub/WWW/Search/960528/cfp.html

Agenda

The Distributed Indexing/Searching Workshop will span two days. The first day's goal is to identify areas for potential standardization through several directed discussion sessions. The second day's goal is to filter the list of issues to identify those most like to lead to useful standards.

The format for sessions during the first day is as follows:

- Two plenary talks (lasting 15 minutes each) expressing opposing views on the session topic including 5 minutes of clarification questions. There will be a strict cutoff of questions based on time and focus (save the discussion for the breakout session).
- A breakout session with three parallel tracks. The goal of the breakout sessions is to examine the potential for standardization efforts in three time periods: 1-3 months, 3-12 months, and 12+ months. A session chair will guide the discussion and prepare a short summary.
- Session summaries from each chair and plenary discussion from workshop attendees.

Although the sessions have been designed to bring out controversy, it is not necessary to determine a *winner*; different approaches may be reasonable over different time frames or circumstances. The goal of the breakout sessions is to identify the issues and to suggest directions for standards efforts.

The first day of the workshop will expose possible directions for standardization efforts in the area of distributed indexing and searching. We have selected three areas based on the position papers submitted.

The first session, distributed data collection, will address issues associated with the collection of data across the network. Is `robots.txt` adequate for future needs? What is the value of protocol- and programatic-based solutions? The topic for the second session is data transfer format. Early deployments may create a dominant standard such as the Virtual Software Library. Format negotiation enables interoperable access to multiple standards. The final session will examine the need for architectures that distribute search across several repositories. The most popular indexes today are constructed as centralized repositories in the mainframe model. More recently, meta-search engines have become more popular. Decentralized, topic-specific indexes take advantage of the restricted domain to add functionality. What is the role of repository access protocols like Z39.50 and other mesh-like models appearing on the Web? Is distributed searching a realistic paradigm for administratively decentralized resources?

The schedule:

- **Session I: Welcome and Overview** 9:00-9:45
  Speakers: Michael Schwartz (@Home) and Mic Bowman (Transarc)
- **Session II: Distributed Data Collection** 10:00-12:00
  Speakers: Darren Hardy (Netscape) and Martijn Koster (AOL/WebCrawler)
  Breakout session chairs: Nick Arnett (Verity), Carl Lagoze (Cornell), Wick Nichols (Microsoft)
- **Lunch** 12:00-1:00
- **Session III: Format Negotiation/Standardization** 1:00-3:00
  Speakers: Pete Lega (c|net) and Luis Gravano (Stanford)
  Breakout session chairs: Gary Adams (Sun), Mike Heffernan (Fulcrum), David Eichmann (Univ. Houston - Clear Lake)
- **Break** 3:00-3:30
- **Session IV: Architecture for Distributed Search** 3:30-5:30
  Speakers: Andrew Van Mil (OpenText) and Dan LaLiberte (NCSA)
  Breakout session chairs: Ray Denenberg (LOC), Clifford Lynch (UCOP), Ken Weiss (UC Davis)
- **BOF I**: Spidering Agreements 5:45-6:45
- **BOF II**: ??? 5:45-6:45

The second day will begin with a brief summary by the workshop chairs. This summary will contain a coherent overview of the efforts from the first day. The first session will discuss and enumerate potential standards directions. During the second session participants will break into groups to formulate written statements for incorporation into the workshop report. The schedule:

- **Session V:** Enumerate Standards Directions 9:00-12:00
- **Lunch** 12:00-1:00
- **Session VI:** Prepare Draft Reports 1:00-5:00

Attendees

Institution	Attendee	Attendee's Email Address
(Independent Consultant)	Cecilia Preston	cecilia@well.com
@Home Network	Michael Schwartz	schwartz@home.net
AOL/WebCrawler	Martijn Koster	m.koster@webcrawler.com
Apache HTTP Server Project	Robert Thau	rst@ai.mit.edu
AT&T	Fred Douglis	douglis@research.att.com
Bellcore	Kshitij Shah	kjshah@ctt.bellcore.com
Blue Angel Technologies	Margaret St. Pierre	saint@bluangel.com
Bunyip Info. Systems, Inc.	Leslie Daigle	leslie@bunyip.com
c|net: the computer network	Pete Lega	petel@cnet.com
CNR-IEI	(repr. by Ralph LeVan)	rrl@oclc.org
Colorado State Univ.	Daniel Dreilinger	dreiling@cs.colostate.edu
Cornell Univ.	Carl Lagoze	lagoze@cs.cornell.edu
Corp. for Nat'l Res. Initiatives	Jeremy Hylton	jhylton@newcnri.cnri.reston.va.us
Crystaliz, Inc.	Sankar Virdhagriswaran	sv@mail.crystaliz.com
Distr. Systems Tech. Centre	Andrew Wood	woody@dstc.edu.au
Excite, Inc.	Mike Frumkin	mfrumkin@excite.com
Fulcrum Technologies, Inc.	Mike Heffernan	Mike.Heffernan@fulcrum.com
General Magic, Inc.	Barry Friedman	barry@genmagic.com
Geodesic Systems	Ellen Spertus	ellens@ai.mit.edu
IBM Corp.	Rong Chang	rong@watson.ibm.com
Index Data	(repr. by Ralph LeVan)	rrl@oclc.org
InfoSeek Corp.	Mike Agostino	mna@infoseek.com
Lexis-Nexis	Chris Buckley	chrisb@chrisb.com
Library of Congress	Ray Denenberg	ray@rden.loc.gov
Los Alamos National Laboratory	Ron Daniel	rdaniel@acl.lanl.gov

Attendees (cont'd)

Institution	Attendee	Attendee's Email Address
Lycos, Inc.	Michael Mauldin	fuzzy@lycos.com
Microsoft Corp.	Wick Nichols	wickn@microsoft.com
MIT and Open Market, Inc.	James O'Toole	otoole@OpenMarket.com
NASA	Rick Borgen	rlborgen@devvax.jpl.nasa.gov
Netscape Commun. Corp.	Darren Hardy	dhardy@netscape.com
Online Computer Library Center	Stuart Weibel	weibel@oclc.org
Open Text Corp.	Andrew Van Mil	ajvanmil@opentext.com
Oracle Corp.	David Robertson	droberts@us.oracle.com
Pica - Centre for Lib. Automation	(repr. by Sonya Finnigan)	sonya@dstc.cs.uq.edu.au
Polytechnic Univ.	Dave Rubin	drubin@quasar.poly.edu
Raytheon Company	Tim Niesen	tmn@swl.msd.ray.com
Stanford Univ.	Luis Gravano	gravano@cs.stanford.edu
Sun Microsystems	Gary Adams	Gary.Adams@Sun.Com
Tele2/SwipNet	Patrik Faltstrom	paf@swip.net
Transarc Corp.	Mic Bowman	mic+@transarc.com
UC Davis	Ken Weiss	krweiss@ucdavis.edu
UC Office of the President	Clifford Lynch	clifford.lynch@ucop.edu
UC San Francisco	John Kunze	jak@ckm.ucsf.edu
Univ. Arizona	Udi Manber	udi@cs.arizona.edu
Univ. Houston - Clear Lake	David Eichmann	eichmann@rbse.jsc.nasa.gov
Univ. Illinois at Urbana-Champaign	Daniel LaLiberte	liberte@ncsa.uiuc.edu
Univ. Tennessee/Knoxville	Shirley Browne	browne@cs.utk.edu
Univ. Washington/MetaCrawler	Erik Selberg	selberg@cs.washington.edu
US Geological Survey	Eliot Christian	echristi@usgs.gov
Verity, Inc.	Nick Arnett	narnett@Verity.COM
Vivid Studios	Christian Mogenson	christian @vivid.com
Xerox Corp.	Hinrich Schuetze	schuetze@parc.xerox.com

Position Papers

(Independent Consultant) Preston Position Paper 7
@Home Network Position Paper 8
AOL/Webcrawler Position Paper 9
Apache HTTP Server Project Position Paper 10
AT&T Position Paper 11
Bellcore Position Paper 12
Blue Angel Technologies Position Paper 13
Bunyip Info. Systems, Inc. Position Paper 14
c|net: the computer network Position Paper 16
CNR-IEI Position Paper 18
Colorado State Univ. Position Paper 19
Cornell Univ. Position Paper 20
Corp. for National Research Initiatives Position Paper 21
Crystaliz, Inc. Position Paper 23
Excite, Inc. Position Paper 24
Fulcrum Technologies, Inc. Position Paper 26
General Magic, Inc. Position Paper 27
Geodesic Systems Position Paper 29
IBM Corp. Position Paper 30
Index Data Position Paper 31
Infoseek Corp. Position Paper 32
Knowledge Systems Position Paper 33
Lexis-Nexis Position Paper 34
Library Of Congress Position Paper 36
Los Alamos National Laboratory Position Paper 37
Lycos, Inc. Position Paper 38
*Microsoft Corp.*Position Paper 40
MIT and Open Market, Inc. Position Paper 41
NASA Position Paper 42
Netscape Commun. Corp. Position Paper 43
*NTT Corp.*Position Paper 44
Online Computer Library Center Position Paper 45
OpenText Corp. Position Paper #1 46
OpenText Corp. Position Paper #2 47
Oracle Corp. Position Paper 51
PICA - Centre for Lib. Automation Position Paper 52
Polytechnic Univ. Position Paper 54
Raytheon Company Position Paper 55
Stanford Univ. Position Paper 56
Sun Microsystems Position Paper #1 57
Sun Microsystems Position Paper #2 59
Sun Microsystems Position Paper #3 61
Tele2/SwipNet Position Paper 63
Transarc Corp. Position Paper 65
UC Davis Position Paper #1 67
UC Davis Position Paper #2 69

UC Office of the President Position Paper 70
UC San Francisco Position Paper 71
Univ. Arizona Position Paper 72
Univ. Houston - Clear Lake Position Paper 73
Univ. Illinois at Urbana-Champaign Position Paper 75
Univ. Queensland Position Paper #1 76
Univ. Queensland Position Paper #2 78
Univ. Tennessee/Knoxville Position Paper #1 80
Univ. Tennessee/Knoxville Position Paper #2 81
Univ. Washington/Metacrawler Position Paper 82
US Geological Survey Paper #1 83
US Geological Survey Position Paper #2 84
Verity, Inc. Position Paper 84
Vivid Studios Position Paper 86
Xerox Corp. Position Paper 87

(Independent Consultant) Preston Position Paper

A position paper for the Distributed Indexing/Searching Workshop

Submitted by Cecilia Preston (cecilia@well.com)

I do not have a web site established where I could give this a URL as requested in the call.

With the explosive growth of networked information systems available it is becoming evident that even in this brave new world, the automatic indexing of large collections of information can be handled somewhat effectively. But, many major issues arise from the simple fact that there are a vast array of domains all with their own vocabulary, and that the same 'string' can represent very different concepts depending on the domain. For example, the Final Jeopardy category this evening was DATES. What DATES: the fruit, a notation of time, or part of the courting ritual in North America?

To make web crawlers and other forms of automatic indexing systems more useful, these larger issues need to be taken into consideration.

1) English (American) is not the only language on the planet. The work of the IAB character set workshop in March of this year will frame some of the issues that can be addressed in Internet standards that should allow for a consistent methods for specifying the language of use.

2) Given #1 above, how can the knowledge of language be incorporated into crawlers etc.? t-h-e in English would most likely be overlooked as having no content, but in French the same three letters t-h-e (given loss of the accent that often occurs) has content. Even in the same language (the DATE example above) context is required to make sense of meaning.

3) Vocabulary is used as a shorthand within a discipline to perform a number of functions a) signify that you are a member of the community b) reduce an entire concept or structure to a word or phrase.

4) Metadata is as specific to a discipline as the vocabulary. Very generalized metadata such as the Dublin Core provides a minimal set of elements which are in almost all networked information systems; such as author (someone or some other legal entity is responsible for the creation of the data, and putting it in this form), a title (the shorthand for the entire work) a date of creation or production, etc. All elements that allow for a quick scan for relevance without pulling the entire object or document.

These and other related issues must be taken into account for the standards that are going to be developed which will allow networked information systems to provide both humans and machines the information being sought, without overloading any system. No one standard or small group of standards will be capable of handling this task. The interoperation of standards developed by the many constituencies who are best positioned to describe their data must occur.

@Home Network Position Paper

Moving Beyond File Retrieval For Distributed Indexing
Michael F. Schwartz, @Home Network

My motivation for co-chairing this workshop was to bring together a cross section of people involved with information server technologies, search technologies, and directory and online services, to discuss where repository interface standards could support better approaches to distributed indexing and searching. Beyond reducing the CPU and network load required for indexing, appropriate repository interface standards could allow the Internet/intranet searching market to grow by removing incompatibilities among current tools and services.

It is not the goal of this workshop to produce a standard; I don't believe it is possible to create a meaningful standard in a room with 50 people. Rather, the workshop will be an opportunity to uncover and discuss areas of mutual concern where standards might gain momentum.

I believe a key step towards establishing appropriate indexing and searching standards is to transcend the current file orientation of indexing. The object-at-a-time nature of HTTP was never designed to support indexing, and using files such as robots.txt is too static and flat a paradigm to support many types of meta data. Web crawlers arose to fill a market demand in an environment that provides no other guaranteed means of collecting information, yet I hope this workshop can establish as a common goal the definition of a collection-oriented, programmatic indexing interface that can be used *in addition* to crawlers.

In Harvest we created a mechanism where indexing data could be extracted before it was transmitted across the network to an indexer, placed into a structured format (SOIF), and transmitted using a compressed streaming protocol that supports incremental updates. I see three important ways that those basic ideas might be shaped into a more encompassing framework. First and foremost, I would like to see the ability to *negotiate* a common query language between a repository and indexer. This would allow components that happen to speak the same language to communicate without an intermediate translation. It would also allow components to communicate using application-specific languages (e.g., utilizing a geo-spatial meta data standard), or using heavier-weight languages than is common in network information retrieval environments (e.g., SQL). Second, it should be possible to retrieve information needed to support the relevance ranking heuristics used by full text indexing systems, in addition to retrieving attribute-value structured meta data. Rather than defining an HTTP "MGET" mechanism, I believe the right approach for this would be the ability to retrieve a remotely generated index -- either by agreeing on a standard index retrieval format, or through an index format negotiation protocol. Third, I would like to see standards for remote query interfaces -- both at the language and the user interface levels. In Harvest we defined a simple generic query language (and implemented mappings to several search engines), but in retrospect that was the least successful aspect of the project: because we chose a "least common denominator" approach it did not support important features like relevance ranking and adjacency operators, and hence that language stood no chance of standardization. The Stanford Digital Library group has more recently taken some steps regarding this difficult problem.

We included "distributed searching" in the set of topics for this workshop, and I'm curious how the participants will rate the importance of this problem. I am aware of some interesting efforts to solve the problems that arise when merging the relevance ranked results of a distributed query, but I question the extent to which people will deploy distributed search services in practice. The problem I see is analogous to why distributed database systems never really caught on: if it is possible to reach agreement about the schema, one might as well just run the database on a large centralized bank of servers -- especially since network bandwidths are improving less quickly than CPU, memory, and disk costs. I *do* believe it will be important to segment global search services into topical or community-focused components, but it's not clear that distributed search is a useful paradigm in such an environment.

AOL/Webcrawler Position Paper

Position Paper for the Distributed Indexing/Searching Workshop

At WebCrawler we are pleased to see W3C take an active interest in the area of resource discovery. We believe that offering Web users an effective search experience requires increasingly more sophisticated information exchange between information providers and indexing systems. Practices to accomplish this will only gain critical mass if they are standardised and backed by the industry as a whole, and W3C could play a catalysing role.

The current generation of Web-wide indexing robots[1] all have to deal primarily with the same issues, which would benefit from increased communication between information providers, indexing services, and end users:

Avoiding indexing "bad" documents

This is partly addressed by the Standard for Robots Exclusion (SRE) [2]. The SRE has got some problems, for which we would like to suggest some solutions.

Finding "good" documents to index

This can be addressed by simple mechanisms based on the SRE or other server-centric mechanisms. On a document level, relationships between documents need to be identified.

Describing documents

The suggestion of a small standard set of meta-data on a document level (using META or LINK tags [3]) is an obvious and effective step we'd welcome. More elaborate rating schemes such as PICS could even address group ratings of resources, but are not readily deployed.

Users searching for documents

While differentiation is important in the marketplace, the user would benefit from standard search mechanisms, such as query language constructs.

Efficient indexing

Finally, mechanisms to aid the mechanics of indexing (such as Harvest) would be beneficial, but are likely to be slow in being deployed world-wide, and warrant separate consideration from the issues above.

We look forward to discussing these and other issues further at the workshop.

Martijn Koster, Software Engineer, WebCrawler

---

[1] http://info.webcrawler.com/mak/projects/robots/robots.html
[2] http://info.webcrawler.com/mak/projects/robots/norobots.html
[3] http://info.webcrawler.com/mak/projects/meta/equiv-harmful.html

Apache HTTP Server Project Position Paper

Delegating control over multiple search engines on a site

The simplest way to configure a search engine for a web site is simply to compute one large full-text index of all the files at the site, and then provide a single interface for visitors. However, there are situations where the simplest approach is not necessarily the best.

For example, consider a server which has three text databases --- two separate databases containing end-user documentation on different releases of the same product, and a third on, say, fly-fishing. If we used the same index files to handle searches on all three databases, users trying to search the product documentation would get information on both releases (which would surely be confusing); to make matters worse, they might get a few fish stories as well.

So, this site would need *multiple indexes* to serve all its clients well. If the search engine is integrated with the web server itself (e.g. by means of a server API, which one might well want to do for reasons of efficiency), this means that the server has to provide the search engine with information about all of the different indexes on the site --- somehow.

The need for delegation

The simplest way to design the search engine for this sort of site is to have a single configuration file, which points to index files and other support material for each of the indexes. However, once again, the simplest way to build the system is not necessarily the best.

The reason has to do with maintainability. Maintainers of each of the text bases will presumably want to alter their configurations from time to time. In the single-configuration-file scenario, this means that each of them would have to edit the single configuration file. By Murphy's law, it is inevitable that sooner or later, someone is going to slip up, and alter the configuration of a database which is not their own.

Indeed, it may be the case that some of the text-datbase providers are not trusted to alter the configuration of databases which they don't own. Ideally, the search engine would provide for this --- that is, it would allow for a webmaster to *delegate* the authority to set up a new search engine for a text database to the maintainers of that database, to specific individuals, and them only.

Delegation in context

Search engines are not the only context in which the need for delegation is important --- indeed, many webservers (of which Apache is one) allow the behavior of most server features (directory indexing, server side includes, error handling, CGI scripts, etc.) to be controlled on a per-directory basis by a file located in the directory in question (and the mechanisms that it uses for the purpose, which I have described elsewhere , could be used by search engines implemented as Apache server extensions).

Apache supports this flexibility because its users have found centralized control, and the attendant possibilities of cross-project interference, can be an insuperable administrative headache. Designers of search engine software may find their own users have similar concerns.

Robert S. Thau

AT&T Position Paper

Support for Temporal HTTP Queries: a Position Paper

Fred Douglis, AT&T Research

There has been much work in the recent past in the area of tracking modifications to data on the Web. Typically, a user's agent polls a list of URLs periodically and compares their timestamps, or checksums when timestamps are not available, to determine what has changed. Examples of this approach include w3new, webwatch (now SurfBot), URL-minder, Netscape SmartMarks, and AIDE.

This approach is not scalable. Even for systems that centralize polling for many users (URL-minder has done this all along, while AIDE has moved to this architecture recently), sending individual requests for each page is an unnecessary waste of network and server resources. This is especially true when data are dynamically generated and have no timestamp to compare, in which case the entire document must be generated and transferred to the agent performing the query, which will then compute a checksum. For slowly changing data, all this work is repeated needlessly.

The HTTP community has recognized that establishing a new connection for each request generates unnecessary traffic and results in poor network utilization (due to slow-start). As HTTP evolves to support more complicated or long-lasting requests, I propose that it support in the protocol and on each compliant server the type of functionality that the agents listed above perform on the client side: given a list of `MULTIPLE` documents on a site, it should return the last modification date and/or checksum for all of them in a single operation.

Alternatively, or in addition, it could support a `REGISTER` command to send back electronic notification (via email or a CGI interface) when a document changes. Some sites do this today, either locally or using a link to URL-minder. In the case of pages that are "expensive" to generate (e.g., queries against a large database), it may also be common for the server to cache the results and use some internal state to know when it is necessary to regenerate the page from scratch. In such cases, frequent requests for the same data place minimal load on the server, but sending the pages over the Internet place an unncessary load on it.

Lastly, an additional level of support that would be of use for tracking changes to documents would be to provide *versioned* data. Versioning allows users to determine not only *when* pages have changed, but also *how* they have changed. While AIDE archives pages on demand to provide this data, server-side access to versions of documents would obviate the need for an external *ad hoc* solution, and avoid duplicating pages unnecessarily.

Note that any of these services can be supported at the CGI level, rather than as part of HTTP itself, as long as there is a standard for how to invoke them. Note also that in addition to these personal agents that may poll daily or weekly, there are search engines that periodically scan the entire web. Their goal is similar: they want to find what new pages exist, and what pages have changed. An HTTP or CGI interface to return information about a list of URLs or about all URLs on a site will enable agents on other hosts to retrieve only the pages that have truly changed.

---

douglis@research.att.com

Last modified: Mon Apr 15 18:18:24 1996

Bellcore Position Paper

Approximating a Single Index by Combining the Results of Distributed Searches

Kshitij Shah
Bell Communications Research
444 Hoes Lane
Piscataway, NJ 08854

Haym Hirsh
Bell Communications Research
445 South Street
Morristown, NJ 07960

Leon Shklar
Rutgers University
Department of Computer Science
Piscataway, NJ 08855

It is becoming increasingly infeasible to maintain a single index for the enormous amount of information on the World Wide Web. Consequently, it has become important to be able to access multiple heterogenous indices in the execution of a single information query. The objective of this work is to project an image of a single index even when queries are actually executed against multiple indices distributed across many machines. Our approach to this problem is to meaningfully merge the documents retrieved when executing the query against multiple indices. The main complexity in doing this is that for most ranked retrieval indexing technologies the scores of retrieved documents only make sense relative to a particular index.

We are exploring three specific approaches to this problem:

1 **Indexing a temporary repository of retrievals:** Here we create a temporary repository from the union of all the documents retrieved using each index. We execute the original query against an index created from this new, temporary repository.

2 **Learning scaling factors:** This method finds query-specific scaling factors to renormalize the scores of the documents retrieved using each index. The union of the results is then sorted using the resulting scores.

3 **Learning ordering rules:** Here we learn query-specific rules for identifying the desired relative order for any pair of retrieved documents. At query-execution time we use these learned rules to order the union of the retrieved results.

Our methods for each assume access to different levels of information about each document:

- those that have access only to the scores of the documents coming from each repository,
- those that have access to each indexing technology's internal document representation (i.e., documents' vector representations), and
- those that have access to the full content of all documents.

To test how well we project an image of a single index we are evaluating our methods by testing the extent to which we retrieve the same results that would be obtained when executing a query against a single index for all documents (the "gold list"). We use two measures for testing how the various experiments perform as compared to the gold list:

- **T correlation**. This is a modified version of Kendall's Tau correlation [1] that measures the rank correlation between the list returned by our various methods and the gold list.
- **Proportion overlap**. This is a measure of the amount of overlap between the results of our various methods and the gold list.

We are evaluating our work using two very different indexing technologies -- Wide Area Information Service (WAIS) [2] and Latent Semantic Indexing (LSI) [3]. Our preliminary results show that for WAIS the best results are obtained when the system has access to the full contents of each document so that our temporary-repository approach can be applied, and for LSI, simply sorting the union of retrieved results is comparable to more sophisticated methods that require much more effort with little gain in performance.

[1] M. Kendall and J. Gibbons, "Rank Correlation Methods", Fifth edition, Oxford University Press, 1996.
[2] B. Kahle and A. Medlar, "An Information System for Corporate Users: Wide Area Information Service", Connexions - The Interoperability Report, 5(11), November 1991.
[3] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Hashman, "Indexing by Latent Semantic Indexing", Journal of the American Society for Information Science, 41(6), 1990.

Footnote: *This is a position paper in response to the CFP for the Distributed Indexing/Searching Workshop to be held at MIT, May 28-29, 1996 Sponsored by the World Wide Web Consortium*

Blue Angel Technologies Position Paper

Z39.50 and Semantic Interoperability

Margaret St. Pierre, Blue Angel Technologies

Searching for information from heterogeneous data sources on the Web is often performed by searching for words in the raw text of documents. Such a practice typically results in a hit-or-miss scenario and the relevancy of the search results is often misleading. Gathering search results from heterogeneous data sources on the Web is also difficult, since results are prepared for display purposes only. This practice makes it virtually impossible to compare and contrast the search results gathered from each data source.

To achieve accuracy and precision in a distributed search, there is a need to establish semantic interoperability among the heterogeneous data sources. This semantic interoperability must be established in both the search criteria and in the retrieved search results (metadata).

Although a universal set of search criteria and metadata could theoretically be developed and applied when searching all data sources, it is unlikely to accommodate the specialized needs of every discipline. Specialized disciplines also need to agree on a standard set of search criteria and retrieved metadata, and assign precise definitions to each. Only then will the level of accuracy and precision of a distributed search be guaranteed.

For example, if a client or user agent were to search for information about a specific historical painting from a set of distributed museum resources, it would include an explicit indication that the search criteria is to be interpreted within the context of the museum discipline. Thus, the search criteria may specify that the usage of a search term be restricted to a specific artist or time period, the context restricted to within a copyright notice, and the authority restricted to a well-known art and architecture thesaurus. The client may also specify that the retrieved metadata also be based on the museum discipline. The requested search results may thus be restricted to retrieving the artist's name, the time period of the painting, the location of the original work, the location of reproductions, and a list of related paintings.

The Z39.50 search and retrieval protocol has been designed for the purposes of achieving semantic interoperability among heterogeneous data sources. Z39.50 has a well-defined and well-developed mechanism for specifying search criteria and for delivering metadata in search results.

Although Z39.50 has predefined a global set of search criteria and metadata, Z39.50 also offers a means for specialized disciplines to define their own set of search criteria and metadata. A number of disciplines have already established agreements, for example:

- Bibliographic Data
- Government Information Resources
- Scientific and Technical Data
- Earth Sciences and Data
- Digital Library Collections
- Museum Information

As new specialized disciplines emerge and evolve over time, Z39.50 provides built-in extensibility designed to handle this growth.

Z39.50 has a mechanism (the Explain facility) for a client or user agent to discover the discipline(s) that a server supports. Thus, a distributed search performed across heterogeneous data sources with a common universe of discourse produces accurate and precise results.

Bunyip Info. Systems, Inc. Position Paper

Distributed Indexing and Searching: A Big Picture

Leslie L. Daigle
<leslie@bunyip.com>
Bunyip Information Systems Inc.

Sandro Mazzucato
<pedro@bunyip.com>
Bunyip Information Systems Inc.

May 1, 1996

To properly address the problem of distributed indexing and searching of Internet resources, the component pieces of the problem must be identified and placed in the picture:

- Distribution
    - cooperative gathering for indexes
    - directed searching (query routing)
- Indexing
    - general purpose searching vs. specific applications
    - determination of relevant index (meta) data
    - creation of this meta data
    - standardized representation of this meta data

The suggested discussion topics in this workshop's call for participation seem to stem from a perspective that is closely focused on extant systems for providing some measure of indexing of resources. Specifically, the focus seems to be on the creation and representation of index data for general purpose resource discovery. A broader perspective must be taken.

Our work with Archie and Digger (based on Whois++) have provided experience with distribution in both areas, and insight into some of the issues faced when generating indexing data. We will focus here on issues of distribution across servers and time.

Distribution across servers -- cooperative indexing

Server load for indexing can be reduced by sharing gathered information amongst indexers. Replication of all or only a portion of the indexing information may be achieved in different ways. The approach used in Archie is to divide the gathering responsibilities among all or a subset of the indexing servers and let them collaborate to perform the replication of information.

Similarly, the information server itself can cooperate in the indexing process -- there are contexts in which, rather than allowing indexers to "pull" the indexing data, the server would be better served by making agreements with indexing services to which the data could be "pushed". One standard that needs to be addressed is the "robots.txt" structure. The knowledge of the type of information is always greater closer to its source, and so it seems natural to have a protocol that allows information providers to give more guidelines to the different robots.

Distribution across time -- responsible gathering

When designing an index data gathering system, the volatility of the information being indexed should be considered. The ideal is to design and build a service that provides the most recent and accurate information to its users. However, frequent accesses to data will not necessarily provide indexing information of greater quality. It may only find data that is often modified, such as daily news. On the other hand, one may not want to index this type of information as it will reside at the location only briefly. A more useful approach might be to create an index of (static) infrastructure information surrounding that volatile data. An extension to the robots.txt convention, specifying the minimal interval between retrievals may help reduce the extra load on information servers.

These examples serve as illustrations of the many and varied issues that must be considered in order to develop truly global indexing systems. The focus of attention at the workshop must be raised beyond the simple creation of a representation for index data, as this is only a small piece of the big picture.

Virtual Software LIBRARY

Virtual Software Library

P. Lega, Z. Turk Ph.D., C. Webster, M. Barzun

The ability to download free software, updates, drivers, patches, and demos of commercial software remains one of the main motives for connecting to the Internet. This is not surprising: Unlike many other types of Internet data (such as news, reviews, video clips, and stock quotes), which all have proven and effective traditional distribution channels to compete with (magazines, TV, radio), software's most efficient distribution channel is clearly the Internet. We can safely expect that in the future the role of the network in the distribution of software will grow and that we will see a dramatic increase in the number of commercial software packages distributed on the Internet as well an increase in the number of independent software authors who will be able to sell directly through the Internet.

Perhaps the most important advantage that the Internet has over other information distribution channels is that is reduces the information overload by offering the just-in-time information delivery (as opposed to just-in-case paradigm we experience in books and magazines). This means that if and when we need a certain kind of information, we go out and ask something to find it for us. The information the user brings into this search can be broken into two parts: 1) who do you ask and 2) how precise is your question. The better information one includes in the question, the more likely the most relevant information is found. Another factor (3) that determines the success of the search is the type of database being searched. For example, a search for a paper on demographic policies in SE Asia is likely to achieve more relevant results if the database searched is one edited by qualified librarians and not simply one consisting of every document on the Web.

Before the development of the VSL (Virtual Software Library) started in 1994, there were two ways to find software on the Internet. The first was to know the file name and then ask Archie which computers that program could be found on. Archie is a system that collects information on which files are stored on what servers. The second way was to go to a server "known" to be specialized in a certain type of software and try to find the file by looking into the catalogue of that site (structured edited information in a proprietary format).

The design goals behind the VSL were (a) to create one engine that could find the most relevant software available on the Internet, (b), to make use of the edited, structured information on the files, which were ready and available on the specialized servers (c) to help make access to those files easy, reliable and fast and (d) to make this in such a way, both technically and organizationally, that any piece of software could appear in the VSL. We assume that the trend toward software being written as components (Java, ActiveX, etc.) will dramatically increase the number of software files distributed over the Internet. We do not believe that a single organization can possibly maintain a high quality catalogue of all that and keep it up to date. On the contrary, we are convinced that there are specialists out there, who do a great job of cataloguing a certain type of software such as games, winsock utilities, ms-dos files, etc. We are sure no one can make a catalogue of IBM software as well as IBM can - but we are also convinced that there should be one place too look both for IBM's and other independent software vendor's products. This makes the VSL conceptually very different from someone who might simply buy a few CD-ROMs and put them on the Web under their own label. The catalogs included in the VSL keep full credit for their work.

The initial information schema of the VSL (Turk, 1995) allowed for such distributed cataloguing, registration, and downloading of software, but centralized the search services. The VSL database has been freely available to several search sites (called VSL Front Desks) on 4 continents. Since the VSL has been taken over by c|net: the computer network in May 1995, the major next step was the publishing of a standard format for the description of software archives (VSL-OF1, the Virtual Software Library Open format 1.0).

Several major software companies such as Adobe, IBM, Intel, Lotus, SGI, and Borland have reorganized their archives so that they could be included in the VSL. Recent developments include the ability to register not only whole archives but single files and richer file level information.
Over the last year of explosive growth, the VSL team gained insight into creating a huge meta-archive from harvesting externally generated index and archive information via the VSL-OF1 standard:

- Most archivists have limited resources locally to create and maintain accurate and useful content data. We have felt a clear need for the meta-archive to provide tools for its members use to assist generating data for the archive system.
- A meta-archive software system must be flexible and able to "adopt" and remotely manage the indices of otherwise non-indexed archives.
- A meta-archive system must support both the tremendous power of terse broadband searching, and the value-added by assembling and editorially enhancing its more popular and timely content.
- Maintaining a consistent interface for using and participating in this system has helped greatly, but the larger challenge has been to provide the supporting automation tools and resources to fuel the archives growth.

We have started solving the problem of poorly described content by implementing tools that allow in-house editorial staff to add value to the harvested information. This feature is key when attempting to present meaningful results to the end-user from sparsely annotated remote archives. This differentiaton from most software search engines makes the VSL more useful to less-technical end users.

A balance of interface standards and flexible modes of archive participation appear to be the key to the success of a broad user base meta-archive.

Call for Papers

CNR-IEI Position Paper

Online Public Access Catalogs and the World Wide Web Maria Bruna Baldacci, CNR-IEI John Favaro, Intecs Sistemi Online Public Access Catalogs (OPACs) are complex systems that must respond to a multiplicity of requirements, often quite sophisticated. For example, they must be able to search for all of the works of a particular author, distinguishing them from those of other authors with the same name; or they must be able to search for a particular edition of a work; or different editions of the same work, even when they are published under different titles. Furthermore, they must deal with information of an administrative nature - for example, recording whether a book is available or out on loan.

For this reason, the bibliographic information managed by an OPAC is augmented not only by a set of administrative data, but also by a complex infrastructure whose purpose is to permit the bibliographic control of the form of names, titles, editions, etc. It is in this way that OPACs distinguish themselves within the broader category of general information retrieval systems

The international standard information retrieval protocol Z39.50 has been developed expressly to allow the user to avoid having to deal with problems associated not only with network connectivity, but also with different user interfaces, different names for elementary bibliographic items, and different languages.

Increasingly we are seeing access to online public access catalogs via the World Wide Web proposed as an alternative to the Z39.50 protocol. But from a bibliographic point of view, there is no comparison between these two approaches. Although the Web might seem easier to use and capable of at least handling relatively simple searches, from a bibliographic viewpoint it is much less satisfactory, principally due to the underlying model of communication in the Web.

In communication between client and server, HTTP plays a role of mere transport, and the browser plays an essentially passive role, limited to reconstructing locally the pages, forms, or applets that arrive from the server. As a consequence, the browser receives bibliographic descriptions in textual form, not as structured records which carry semantic information that can be utilized in other applications, and which allow visualization of the data in various forms that may be different from the form in which they are originally received from the server.

But the severest deficiencies become evident directly in the searching process. First of all, the problem of diverse user interfaces is not resolved. The consequences are perhaps not so evident in a national context, because the OPACs that are accessible via the Web within a single country tend to have fairly similar home pages, and thereby might give the impression that it could be possible to have a homogeneous form of dialogue. But even in national contexts, this impression is erroneous: although it is true that the "classic" OPAC access points are presented with easily identifiable names in all of the forms, it is also true that the access points of the individual OPACs may have very different access methods associated with them, and the user cannot be sure that the access methods he has in mind while creating his query (for example, access via string versus access via keyword) are really those associated with the access points he has chosen. This situation frequently results in search failure.

Z39.50, on the other hand, has been conceived to address directly the requirements of the user-system dialogue. Through the EXPLAIN facility and the SCAN service, Z39.50 allows the client to learn the characteristics of the server, the structure of its databases, the access points of each database, and the access methods associated with each access point. The application-level nature of Z39.50 enables the client to configure itself with all of the information known to the server and guide the user through the dialogue specified by the associated OPAC, yet always presenting the user with a uniform interface.

In summary, the Web can provide attractive solutions to many problems of information access and indexing, but the bibliographic domain requires the full-fledged application level capabilities that are embodied in the Z39.50 protocol.

Colorado State Univ. Position Paper

Formal Standards for Meta-Search

Daniel Dreilinger

May 6, 1996

Meta-search engines, tools that simultaneously search multiple conventional search engines and integrate the results, are becoming increasingly popular. Today at least three meta-search engines are in wide use: SavvySearch, MetaCrawler and WebCompass; many more are in development. Users report that these tools are very helpful in their Web navigation endeavors. In some cases, the search engines that are queried by meta-search engines find this behavior beneficial. Lesser known search engines enjoy the publicity and extra awareness that the meta-search tools raise. Meta-search engines also serve as additional entrances into sites whose search engines index local content only.

In other cases, as recently suggested on the robots mailing list, meta-search engines appear to work against the advertiser supported business model adopted by some of the larger search sites. Related problems that have surfaced are the increased strain on the Internet and various search engines, and reformatting of results. One solution to the advertising problem that has been suggested is propagation of advertisements produced by search engines into the meta-search results. Another solution might involve intermediate result pages which give search engines an opportunity to display advertisements for each of their links that is followed.

Ultimately it should be up to search engine providers to decide how and under what conditions their resources are used, and each will probably have a unique opinion. Perhaps these problems are best addressed with the introduction of a formal standard for meta-search tools. A standard for meta-searchers could exist as an extension to the existing robot exclusion standard, or as an entirely new mechanism (how about SavvyNotWanted.txt?) Below is a partial list of questions that I believe should be considered when designing such a standard:

- Where and when are meta-search agents welcome? (i.e., certain peak hours that should be avoided.)
- Are there maximum resource quotas that should be observed? (i.e., maximum allowable number of queries per day.)
- How much liberty may be taken in reformatting results? (i.e., bypassing or changing format of advertisements.)
- Are there other special instructions meta-search designers should follow?
- Is there a protocol for searching a fee-based service on behalf of a registered user?
- How can we avoid infinite cycles of meta-searcher queries?
- Does the standard need to be machine parsable?

This list has probably overlooked some important issues. The next step is consulting the many search engine providers and identifying their concerns.

Cornell Univ. Position Paper

An Architecture for Aggregating Distributed Data and Meta-Data Objects

Position Paper for W3C Distributed Indexing/Searching Workshop

Carl Lagoze, Cornell University, Computer Science Department- lagoze@cs.cornell.edu

The Cornell Digital Library Research Group has been researching and developing protocols and architectures for distributed digital libraries. One result of this work has been Dienst, a protocol and reference implementation for distributed multi-format document libraries. Dienst is the technical foundation for the Networked Computer Science Technical Reports Library (NCSTRL), a collaborative effort by a number of universities to make their computer science technical reports and other relevant materials available to the public.

Our model for distributed indexing in the current implementation of Dienst is rather simple. Each document is stored in a *repository server* as a uniquely-named (with handles) aggregation of meta-data (cataloging data) and one or more representations, or formats, of the document. Each repository has an associated *index server* that scans the bibliographic entries for the documents, indexes the contents, and responds to simple bibliographic searches. User search, browsing, and retrieval access to the distributed collection is provided by a set of *user interface servers*, which broadcast a user search to individual index servers and then combine the returned results sets into a uniform hit list. A set of backup index servers, which periodically gather indexing information from the distributed indexes, provide a level of fault tolerance for the distributed indexing process.

Our current work is focused on near-term enhancements to the existing Dienst architecture and, for the longer term, the definition of a more flexible and extensible repository and indexing infrastructure. Our near-term goal is to move from the current broadcast search model to a selective broadcast model. In this model, a user query is first filtered through a large granularity search, which returns a set of index servers, to which a fine granularity (document level) search is submitted. We are examining two technologies, GlOSS and Harvest, as the means of achieving this goal.

Our longer term infrastructure work is based on the Kahn/Wilensky Framework for digital library objects. We are motivated by a number of requirements related to meta-data. An individual digital library object may have multiple packages of meta-data associated with it. For example, the meta-data for an object may be a combination of a full MARC record; a simple cataloging record such as that represented by the Dublin Core; a description of the terms and conditions describing the rules of access to the object; and the like. Each of these meta-data packages may themselves be library objects, with associated meta-data. A meta-data object may be shared by several digital library objects. For example, the set of objects in a single repository may share a single terms and conditions meta-data object. Furthermore, a meta-data object associated with a digital library object may reside in a separate repository. Finally, there is the need to accommodate complex meta-data types (*e.g.,* Java applets), such as those required to mediated complex terms and conditions.

In response to these requirements we are examining and prototyping a container architecture for aggregating related digital library objects. The container architecture is recursive, in that objects within containers may themselves be containers; distributed, in that objects may be indirectly referenced from containers using URN's; and extensible, in that objects are strongly typed within a CORBA-based distributed object framework and the type system can be extended through the a CORBA-like type registry. We plan to use this architecture as the basis for the next generation of Dienst and as a foundation for future research in distributed indexing, object replication, searching over heterogeneous meta-data, and methods for expressing terms and conditions to protect intellectual property. aggregating related digital library objects. The container architecture is *recursive*, in that objects within containers may themselves be containers; *distributed*, in that objects may be indirectly referenced from containers using URN's; and *extensible*, in that objects are strongly typed within a CORBA-based distributed object framework and the type system can be extended through the a CORBA-like type registry. We plan to use this architecture as the basis for the next generation of Dienst and as a foundation for future research in distributed indexing, object replication, searching over heterogeneous meta-data, and methods for expressing terms and conditions to protect intellectual property.

Corp. for National Research Initiatives Position Paper

Creating Collections with a Distributed Indexing Infrastructure

Position statement for Distributed Indexing/Searching Workshop

Jeremy Hylton, Corp. for National Research Initiatives

This note suggests two ideas that may be useful in the construction of an infrastructure for distributed indexing, searching, and browsing systems. First, I make a clear distinction between servers, which provide storage for digital objects, and collections, which organize related documents. Second, I argue that multiple, independent indexing systems may each require access to the original documents.

These ideas do not lead directly to suggestions for standards, nor are they completely original. They do offer a different perspective on the problem and place different constraints on developing standards.

Distributed information retrieval is often based on a model where many independent servers index local document collections and a directory server (or servers) guides users towards the independent indexes. This model assumes that the documents stored at a particular location define a collection.

In traditional information retrieval, term weights for a document are assigned using a collection-wide statistics, e.g. words occuring in only a few documents are weighted more heavily. This collection-wide information (term due to Viles and French [4]) greatly increases effectiveness and enables other useful services, like automatically constructing hierarchies with scatter/gather [2] or helping users re-formulate queries (content routing [3]).

Applying traditional term weight strategies in a distributed system is hard, because the definition of "collection-wide" can be difficult to pin down and when it is collecting the information can be expensive.

Sheldon [3] proposes a distributed IR model with the important characteristic that a collection of documents is described by a content label and the content label can itself be treated as a document and included in another collection. Content labels help users manage and explore very large information spaces, but the idea could be valuably extended by treating collections (and their labels) seperarely from servers. Thus, a collection could include particular documents from many servers. (HyPursuit [5] moves in this direction.)

Consider a simple example: Several newspapers provide servers with their articles. We could construct many collections, each with different term weightings -- business articles from each of the newspapers, articles with a San Jose dateline, or movie reviews. Different terms would be useful in each collection.

Recent work in distributed indexing has focused mostly on efficient indexing -- minimizing load on servers and keeping indexes small. This is accomplished in part by indexing surrogate for documents that includes only part of the text (in Harvest, the first 100 lines of text and the first line of later paragraphs).

There is a tension between efficient indexing and the collection-based indexing; the best choice of indexing in general isn't necessarily the best for any specific case. An indexing surrogate may omit important terms that occur late in the document or mis-represent the frequency of particular terms.

We can address this tension, in part, by creating a more flexible infrastructure that allows multiple indexing schemes to access to the full content of documents they are indexing. Where a Harvest gatherer describes a single surrogate for a document, a more flexible gatherer would generate surrogates according to a particular index's specifications.

Ideally, the system should be flexible enough to allow very different indexing schemes, including indexes that include word proximity information, n-gram based approaches that don't focus on words per see, or knowledge-based or natural language processing approaches. One possibility is for indexes to send the gatherer a program for generating document surrogates. The gatherer could run the program and return the results to the index.

References

- 1. C. Mic Bowman, et al. The harvest information discovery and access system. In Proc. of the 2nd World-Wide Web Conf., Chicago, December 1994.
- 2. D. Cutting, et al. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In Proc. of SIGIR '92. Copenhagen, Denmark, June 1992.
- 3. M. Sheldon. Content Rouing: A Scalable Architecture for Network-Based Information Discovery. PhD thesis, MIT Dept. of EECS, Oct. 1995.
- 4. C. Viles and J. French. Dissemination of Collection Wide Information in a Distributed Information Retrieval System. In Proc. of SIGIR '95. Seattle, Washington, July, 1995.
- 5. R. Weiss, et al. HyPursuit: A Hierarchical Network Search Engine that Explotes Content-Link Hypertext Clustering. Proceedings of Hypertext '96, Washington, DC, March 1996.

Crystaliz, Inc. Position Paper

Why We Need To Index Objects, Not Documents and Why "Meta" Should Refer to a Language Not a Protocol

Sankar Virdhagriswaran, Crystaliz, Inc., 696 Virginia Road, Concord, MA 01742

The cover material for the workshop shows the limitation of the current thinking in supporting resource discovery on the Web. While the workshop cover brings up and discusses important problems in the discovery of documents - a short term problem, it does not look at the problem for the long term - the web as the place to find and use *any resource*. In this position paper, we argue that the indexing schemes for the future web should consider two perspectives: *Object Discovery* and *Meta Object Protocol*. Our comments are based on our experience in the manufacturing sector and meta object protocols.

First, while the web has always been envisioned as an infrastructure where objects of various types (documents produced by humans being one of the types) can be found and used, the indexing schemes used today (e.g., www.lycos.com) are manifestly insufficient for objects which can be and have been formally described. If one considers examples such as manufacturing parts which can be categorized and described using Group Technology, engineering design documents which can be described using PDES/STEP standards, one realizes that many industries across the world use semantic indexing schemes that have been perfected for their use. Furthermore, services that are being standardized in the Object Management Group such as the trading service are an attempt by object vendors to support wide area discovery of program objects. Finding and using such objects is an important aspect of resource discovery in the WWW.

Second, one-size-fits-all approach to describing the semantics of meta information that can be exchanged between various repositories (such as the Dublin meta protocol) simply will not be used except for documents. Consider one of the fields in the Dublin meta protocol: *language.* If I write a Java applet what value should this language field contain - the Java language, Java VM code, or the native code generated by my Symantec compiler on my Intel machine or all of the above. Each of these uses of the keyword *language* are very important to a person trying to discover a Java applet. However, since the Dublin meta protocol considers human written documents, it does not allow for these nuances. In the long run, what will work is an approach the delivers a mechanism by which the meta information can be described consistently and the set of described entities extended gracefully. Putting the specifics of one set of semantics before thinking about a way to describe the semantics is a short term fix which will only proliferate the *meta* protocol standards achieving no convergence. What is needed is a *meta object protocol* - i.e., a formal way to describe objects about objects.

These observations are based on work that we are performing at Crystaliz, Inc. using Knowledge Query Manipulation Language, PDES/STEP part library standards, and standards work proceeding at Object Management Group. During the workshop we would like to present our experiences based on our efforts in the *formal objects* part of the web.

Excite, Inc. Position Paper

Additions to the robots.txt Standard

Introduction

The robots.txt standard is a very useful tool for both webmasters and the people who run web crawlers. This standard could be even more useful with several additions. The additions suggested below were inspired both by comments from webmasters and by front-line experience developing and running the Excite web crawler.

Site naming

Site naming poses several problems for maintainers of web indexes. Sites can be referenced by many names, and it can be hard to determine which name the webmaster prefers. Also, large sites can be referenced by many different physical IP addresses.

Multiple names

Most sites can be referenced by several names. To avoid duplication crawlers usually canonicalize these names by converting them to ip addresses. When presenting the results of a search, it is desirable to use the name instead of the ip address. Sometimes it is obvious which of several names to use (e.g. the one that starts with www), but in many cases it is not. The robots.txt file should have an entry that states the preferred name for the site.

Multiple IP addresses

Many high traffic sites use multiple servers. Machines are added frequently and their ip addresses often change. Crawlers do not have a good inexpensive way to understand and keep track of the everchanging mapping of servers to logical sites. This causes needless duplication of effort by the crawler and higher traffic at the sites. The robots.txt file is an ideal place to include a list of ip addresses that map to a logical site.

Freshness of content

HTTP provides mechanisms for determining how recently a file has been modified; it even provides mechanisms for avoiding data transfer costs if the file has not changed since the last visit of a browser or crawler. However, the performance of both crawlers and the sites they visit could be improved by providing higher-level information about when content on a site has changed.

Freshness of web pages

One addition that could dramatically reduce traffic would be a representation of modified dates for various parts of the site. Today the only way to tell what pages you want to update is to use the If-Modified-Since request-header field. This costs a connection per page. Having this information centralized in the robots.txt file would decrease server loads. This information could be presented at a directory or file level depending on the size of the site and the granularity of information the webmaster wants to present. A useful representation might be a reverse-chronological list of files and the dates that they were last modified.

Freshness of the robots.txt file

The robots.txt file needs to include a time-to-live (TTL) value. This tells crawlers how often they should update the robots.txt information for that site. Some sites very rarely change their robots.txt files and do not want the extra traffic of having them frequently re-read by multiple crawlers. Even if the If-Modified-Since request-header field is used, a connection still has to be created each time. On the other hand, some sites change their robots.txt files regularly and often. They are often hurt by extensive caching of robots.txt information by crawlers. Having an explicit TTL value would help crawlers satisfy each site's requirements.

Flexibility of the robots.txt file

Although the simplicity of the robots.txt file is a benefit, many sites on the internet today have structures that are too complex to represent with the current robots.txt format.

Multiple content providers

In some instances many people might provide the content for a single site. A good example is a university site which has a separate area for each student. Each of these individuals might want to control access to his or her own section of the site. It is often unreasonable to allow all of the individuals to edit one global robots.txt file. The robots.txt file should have a way to redirect the crawler to read separate robots.txt files from further down in the site. This allows different robots.txt information to be specified for separate parts of the site.

Complex directory structures

The disallow statement of the current robots.txt standard could be made more powerful. For various reasons some sites cannot change their on-disk layout and may have very large directories. It is very cumbersome to exclude part of a large directory using the current disallow statement. A more powerful regular expression syntax or an 'allow' directive to override the disallow for specific files would be useful.

Description of the site

An optional description of the site would also be a welcome addition to the robots.txt standard. A brief human-readable statement about the site's purpose and the kind of content it contains would provide useful information to the end users of the various repositories created by web crawlers.

Conclusion

We have described several improvements to the robots.txt standard. These would improve the performance and usefulness of both web crawlers and web sites. We have not provided details on what the changes to the format should look like but none of the improvements seem difficult to specify.

Mike Frumkin (mfrumkin@excite.com)
Graham Spencer (gspencer@excite.com)
Excite Inc.

Fulcrum Technologies, Inc. Position Paper

World Wide Web Consortium Distributed Indexing/Searching Workshop

Mike Heffernan, Glen Seeds

In response to the Call for Participation for the above workshop, Fulcrum Technologies Inc. submits for discussion position on the issues requested for comment.

Robot Exclusion, Web Crawling and Web Indexing

The general approach of "web crawling" - that of unleashing a software program to periodically retrieve and index the contents of the Internet into a central searchable collection - is an inherently unscalable.

A far stronger approach is to avoid large central indexes, and adopt a strategy of distributed indexing and searching.

Distributed Indexing and Distributed Searching

Fulcrum support the continued evolution of standard protocols (like z39.50) and query syntax that enable distributed search applications in an immediate response model.

Agent based search distribution is an important long term technique. Agents should traverse a set of distributed indexes, as opposed to executing a direct web crawl. Agent based queries will ultimately be more effective, as they do not need to return immediately with a quick result.

The significant challenges in introducing distributed indexing and distributed searching are not necessarily technical in nature. Large Web indexes are now corporate assets that require significant investment in equipment, software and bandwidth to construct and maintain. In order for distributed indexing to succeed, some consideration will need to be given to the business model.

Matching Semantics of Document Properties and Meta-data

There must be a common syntax for recording meta-data and a common semantic ontology for interpreting it. Fulcrum supports the development of a common syntax and basic ontology to allow for rudimentary meta-data extraction from data sources and document collections.

In order for this development to be practical and timely, it will be necessary to limit its scope. Some of the current efforts to capture an web oriented ontology, like Yahoo, could serve as the foundation for this effort.

Merging Result List Scores

Fulcrum supports the development of a clear multi-vendor standard to allow document index collections to compare and rationalize the statistical weight of a query against the collection as a whole. The resulting collection weighting should be engineered to be a viable mechanism for subsequent normalization of the individual document relevance scores presented to the user.

General Magic, Inc. Position Paper

Distributed Web Indexing Using Telescript

Rory Ward, Telescript Bard, rory_ward@genmagic.com
Barry Friedman, Code Dependent, barry@genmagic.com
General Magic Inc: http://www.genmagic.com
Submitted to the W3C Workshop on Distributed Indexing/Searching.

Web Indexing Using Telescript

Telescript is an open, object-oriented, remote programming language. It is a platform that enables the creation of active, distributed network applications. There are three simple concepts to the language: agents, places and "go". Agents "go" to places, where they interact with the place or other agents to get work done on a user's behalf. Agents are in fact mobile programs capable of transporting themselves from place to place in a Telescript network. Telescript contains mechanisms to ensure the safety and security of mobile programs and the servers that host them.

Telescript documentation and software is available at General Magic's web site. General Magic is proposing a *Common Agent Platform* that would allow the development of interoperable agents using a variety of programming languages. See the position paper at http://www.genmagic.com/internet/cap/w3c-paper.htm for more information.

Telescript technology offers the ability to distribute indexing to the sites where the content resides. A search engine operator can develop an agent that can travel to the content site, build an index that meets their needs, and return with the index information.

Each Telescripted web site implements an open, standardized programmatic interface for agents to access local web content. The resident agents do their work autonomously via this API. Their resource consumption and content access is controlled by the site using facilities provided in the Telescript language. Any indexing agent developer can take advantage of the programmatic facilities at Telescripted web sites.

An indexing agent can remain at the web site that is hosting it, and send update agents to its origin when the content is changed (modified, added or removed). The site operator can control the the rate and size of updates.

As Telescripted Web sites become available, they advertise themselves to a central directory. The directory can be used by agent developers to dynamically discover new Telescripted Web sites.

Advantages of this approach

- **Scales very well:** The resources needed to index web content is spread between the content providers.
- **Always up to date:** Agents stay at web sites and update the central repository when local content changes. These updates can be scheduled by the agent developer or by the web site operator.

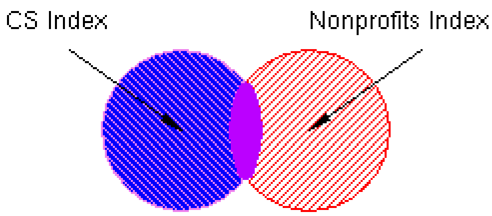
- **Only modified content is reindexed:** Because agents remain local to the content, they only reindex pages that have changed. Therefore, after sending the initial index, only deltas need to be transmitted.
- **Decreased load on the Internet:** Because agents are close to the resources they need, there is no need to transmit every page across the network. Only the index data needed by the agent's developer needs to get transmitted.
- **Robot access differentiated from browsing:** Because indexers are not using HTTP to index the web, robot access can be differentiated from browsing. This helps the site operators and their advertisers.
- **Site configurable and controllable:** The site operator can control the resources used by agents and the content accessed. The site can restrict who can send agents to the site and what security policies are applied.
- **Open platform for indexing:** Any index agent can take advantage of the programmatic interface available at a web site. This allows for different indexing algorithms to be employed.
- **Dynamic discovery of Active Web Sites:** A central directory of active web sites that is always up to date allows indexing agents to dynamically discover new sites.
- **Avoids copyright issues:** Accomplishing the indexing on the local server under the web site owner's control avoids copyright violations and related liabilities for the crawler operator.
- **Multiple varieties of architectures:** As personal web servers and other kinds of "low-level" sites proliferate and become ubiquitous, it seems likely that indexing hierarchies will arise. Using Telescript agents inherently allows for the kinds of flexibility necessary to effect and maintain such a distributed autonomous hierarchical architecture.

Geodesic Systems Position Paper

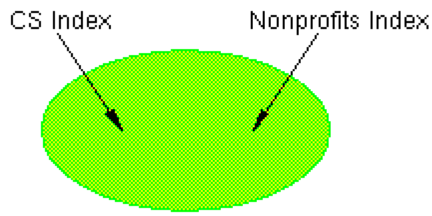
Ellen Spertus (MIT/UW) and Gregory Lauckhart (UW)

Link Geometry

A limitation of current search engines is that they only make use of the text on pages. This ignores the information encoded in the links among pages. Consider the set of pages within three links of a computer science index. The pages within one link of the index are almost certainly related to computer science; the pages two or three links out have a lower probability, although still greater than that of random pages on the web. We can think of the set of pages within *n* links of an index *I* as being the interior of a circle (hypersphere) of radius *n*. We could create two such circles with centers (foci) representing different subject indices and intersect them in an attempt to find material that is relevant to both subjects.



(a) Intersecting circles



(b) An ellipse

Intersection is probably not the ideal operation, since it could exclude a page very close to one focus but a little too far from the other, while including a page the maximum distance from each focus. Instead, we probably want to take the set of points where the sum of their distances from the foci is less than a constant; in other words, an ellipse.* We used our system to find empirically that paths from Yahoo's CS and public-interest indices met, appropriately, at Electronic Privacy Information Center. While we have defined the distance between two pages as the shortest path between them, other possible definitions would take into account the number of paths between two pages.

Applications

Search engines could maintain databases with link information, or engines could perform crawling on demand. For example, to look for physics humor, a search for ``physics'' can be done on the pages reachable from Yahoo's ``humor'' page or vice versa. (Yahoo does allow searching for a piece of text in a subtree, but that only searches the titles and descriptions of the pages stored at Yahoo, not the full text.) Alternately, nodes could be expanded from each of the two Yahoo pages until an intersecting node was found.

Crawling on demand also allows finding information that has not yet been indexed by other crawlers. For example, when searching for information on Cecilia Rodriguez, we were given a URI for a page that no longer existed. We removed text from the right end of the URI until we found a page that still existed. By expanding links from that page, we were able to find not only the moved page but also information on Cecilia Rodriguez that had not yet been indexed. Such searches can be facilitated by preferentially expanding pages or links containing relevant text.

Another application of link geometry is discovering the class of a page from its link structure. For example, a homepage would tend to contain links downward in the file hierarchy with perhaps a few upwards and sideways, while an index would include a large number of links to many different sites. A clue to the category of a page can be found by walking backwards up links that point to it until the first Yahoo page, for example, is reached and looking at the header. Another application of back links is automatically finding pages (indices) that point to pages you like; expanding from the index may help you discover more such pages.

IBM Corp. Position Paper

Distributed Data-Gathering for Web Search

E. Brown, R. Chang, H. Chong, J. Prager, and E. So

IBM T. J. Watson Research Center
P. O. Box 704
Yorktown Heights, NY 10598

{brown, rong, herbie, prager, edso}@watson.ibm.com
Proceedings of the W3C Distributed Indexing/Searching Workshop

Due to the size and growth-rate of the Web, a good distributed indexing/searching mechanism must be integrated with a distributed data-gathering mechanism. Traditionally, this gathering is done by means of a web-crawler. Now, absent a notification protocol in HTTP, the crawler must look everywhere to get the latest data, and since many web pages change frequently, this means the crawler must be continually active. This poses burdens on remote servers and the network itself, and is compounded by the fact that many crawlers are simultaneously trying to do the same thing.

Our investigations suggest that an approach similar to Harvest's use of Gatherers and Brokers is required, but with more generality. In particular, the SOIF usage needs to be extended to accommodate link information and hierarchical representations. If this is done, then a Harvest-like system can interoperate with arbitrary web crawlers by producing standard sets of output files. One such file would be an associated configuration file, used to describe the location, format, date and content of the other files. These files may be altogether absent if the site administration does not want to participate, or may only contain a subset of the public domain if the administration deems its complement to be not useful to crawlers. This methodology turns the essentially confrontational 'robots.txt' approach into a collaborative one where everybody wins.

In addition to the files representing the text, other files such as lists of outgoing links also need to be generated, to supply a complete functional replacement to a web-crawler visiting every page at a site. If a site is unwilling to provide all this data, then at a minimum a file enumerating all of the site's URLs, along with last-modified dates, could be used to advantage (some FTP sites already do something similar).

A serious issue, not yet resolved, is exactly what format the text output should take. It could be a representation of the web pages, which in turn raises the question of whether it should be keywords only, full-text, full-text plus tags, or full-text augmented by results of name-finding and related processes. Alternatively, it could be an inverted index, which raises questions of both format and content. No single solution to these questions will satisfy everybody, in part because different sites will be willing and able to devote different resources to generating and maintaining these output files. It is suggested that a variety of standard levels of detail be established, and the aforementioned configuration file be used to describe the choices made.

Index Data Position Paper

Z39.50 and the World Wide Web

Sebastian Hammer, Index Data
John Favaro, Intecs Sistemi

The tremendous success of the World Wide Web, and the increasing use of WWW front-ends to library catalogs and other information systems has caused decision- makers to question whether the investments required to establish additional Z39.50 services are still warranted. Meanwhile, the increased versatility of the 1995-version of the Z39.50 protocol - which enables it to provide powerful services outside of the strictly bibliographic application domain - leads information specialists to wonder where the WWW and Z39.50 fit together in the evolving information infrastructure.

The Web is an ideal vehicle for organizations that are "vertically integrated," that is, which are owners of content that they can present to the user in a structure of their own choosing. That is why many media and entertainment companies are showing a great interest in the Web today. But when users must actively search the Web for information across organizations, they encounter a sea of largely unstructured data.

The library community has much to offer in the way of providing structure to information resources on the Internet. The Z39.50 standard is a concrete representation of this fact. Currently the search engines and indexes of Web resources suffer from the same weaknesses as the interfaces to library systems. No two are alike, and there is no general way to make structured use of the data that they return. With the current growth of the Web, the search engines are becoming increasingly important - a significant portion of the Web community now spends more time looking at search engine output than on any other type of Web page. However, it may eventually become impossible for any one organization to index everything in a useful way. We will need more well-structured access methods to allow searching across multiple indices. Here the power of Z39.50 as a true, mature information retrieval protocol becomes evident.

Z39.50 specifies an abstract information system with a rich set of facilities for searching, retrieving records, browsing term lists, etc. At the server side, this abstract system is mapped onto the interface of whatever specific database management system is being used. The communication taking place between the server and the client application is precisely defined. The client application is unaware of the implementation details of the software hiding behind the network interface, and it can access any type of database through the same, well-defined network protocol. On the client side, the abstract information system is mapped back onto an interface which can be tailored to the unique requirements of each user: a high-school student may require a simple, graphical interface with limited functionality, while an information specialist may need a complex, highly configurable information retrieval engine. Finally, casual users may prefer an interface which blends in smoothly with their word processor, database software, or, indeed, WWW browser.

In summary, the essential power of Z39.50 is that it allows diverse information resources to look and act the same to the individual user. At the same time, it allows each information system to assume a different interface for every user, perfectly suited to his or her particular needs.

Infoseek Corp. Position Paper

Infoseek's approach to distributed search

by Steve Kirsch, *Infoseek Corporation*, *stk@infoseek.com*
Presented at the Distributed Indexing/Searching Workshop sponsored by W3C.

Abstract This paper describes how Infoseek is approaching the problem of distributed search and retrieval on the Internet.

WWW master site list The Comprehensive List of Sites was not available at the time this paper was written (May 15). We need a reliable and complete list of all WWW sites that robots can retrieve. The list should also be searchable by people using a fielded search and include basic contact information. Infoseek would be happy to host such a service as a public service.

Additional robots files needed In order to minimize net traffic caused by robots and increase the currency of data indexed, we propose that each WWW site create a "robots1.txt" file containing a list of all files modified within the last 24 hours that a robot would be interested in indexing, e.g., the output from:

```
(cd $SERVER_ROOT; find . -mtime -1 -print >robots1.txt)
```

In addition, a "robots7.txt", "robots30.txt", and "robots0.txt" should also be created by a cron script on a daily basis. The 7 and 30 files are for the last 7 and 30 days respectively; the robots0.txt file would have the complete list of all files indexable from this website (including all isolated files). This proposal has the advantage of ease of installation (in most cases, a few simple crontab entries) and compatibility with all existing WWW servers.

Collection identification Infoseek's new full text indexing software (Ultraseek) creates a sophisticated fingerprint file during the indexing process. This fingerprint file can be adjusted by the user to contain every word and multi-word phrase from the original corpus as well as a score for each word and phrase. The user can set a threshold of significance as well for more concise output. Similarly, a requestor of the fingerprint file could set a similar threshold, but this would require a more sophisticated interface than HTTP or FTP. Ultraseek is capable of running a user's query against a meta-index of fingerprint files to determine with excellent precision, a rank ordered list of the best collections to run the query against. No manual indexing is required for each collection. Once the system has been stabilized, we will make the data formats publicly available.

Fusion of search results from heterogenous servers Ultraseek performs query results merging from distributed collections in a unique way. We allow each search engine to handle the query using the most appropriate scoring algorithms. The resulting DocIDs are returned to the user, along with a few fundamental statistics about each of the top ranked documents. This allows the documents to be precisely re-scored at the user's workstation using a consistent scoring algorithm. It is very efficient (and IDF collection pass is not required), heterogenous search engines are supported (e.g., Verity and PLS), and most importantly, a document's score is completely independent of the collection statistics and search engine used. Once the fundamental statistics have stabilized, we will make the statistics spec and protocol publicly available. We currently plan to use ILU to communicate between servers.

Knowledge Systems Position Paper

PetaPlex Project

The PetaPlex Project is a project funded by the US Intelligence Community to develop feasible architectures for very large-scale digital libraries -- to meet the future needs of the community and those of large-scale commercial applications. The specific goals targeted in the current phase of the project is to develop an architecture capable of scaling to 20 petabytes on-line with subsecond response time to access random, fine-grained URN-specified objects, at a sustained rate in excess of 30 million tranactions per second.

The current statement of work calls for integrating one million, 20 Gb disks into a coherent system that can attain these performance objectives --- at acceptable cost. To achieve this level of throughput, the current prototype resolves URN's -- finds, fetches, and displays/executes -- in a single packet round-trip and a single seek. To achieve cost feasbility, the architecture is "massively simple" -- it consists only of simple, commodity-cost, COTS technologies that enable near-automatic construction and maintenance of the system.

A principal part of the architecture involves the full-text search of the hypermedia-structured database for many concurrent searches, on the order of 100,000 on-going searches at any time. The scheme being explored is highly-parallelized, both for the incremental maintenance of the indexes, conducting searches, and storing results in persistent and accessible form.

Lexis-Nexis Position Paper

POSITION PAPER: Z39.50 & Ranked Searching

Co-Authors: Dr. Chris Buckley, Chief Scientist, Sabir Research; Peter Ryall, Senior Architect, LEXIS-NEXIS

Access the Distributed Indexing/Searching Workshop Call for Papers using the Workshop URL.

Abstract

In the current universe of relevancy-based search & retrieval systems, there is a wide diversity of search methodologies, ranging from simple term occurrence/proximity algorithms, to modal, LSI, & connectionist logic, to full natural language processing. Across this spectrum there are many variations in query syntax, & in the degree of control given to the user and/or client over the exactness of the interpretation of search terms, as well as over the precision & comprehensiveness of the results selected from the target collection(s). Similarly, within the WWW community, a range of syntaxes exist for input of search query terms & criteria (various flavors of structured forms, fields allowing free-form query text, etc.)

The `Type 102 Ranked Query' currently under development for use within the Z39.50 Search & Retrieval protocol has been specifically designed to accommodate the ranked search technologies used by the majority of large-scale commercial information providers and Information Retrieval (IR) software vendors. The set of features specified within the standardized syntax of the Ranked Query is estimated to encompass the functionality supported by 80-90% of mainstream commercial ranked search technologies (including those in wide use across the WWW).

How the Z39.50 Ranked Query Facilitates Distributed Searching

Using the standardized Ranked Query, a consistent query & search term syntax can be used to send searches to multiple search systems, based on the following key elements of the Query:

- A standardized methodology & an absolute scale for ranking results from a single search server, or from multiple servers (which may use a wide range of different search technologies), ensures that:
  - Ranked results are more consistent & predictable from one system to another;
- A standardized syntax for allowing the user/client to specify search criteria within the query allows:
  - Multiple systems to be searched consistently & concurrently;
- Standardized methods are supported for combining ranked results from disparate search systems, making the Ranked Query very powerful in a distributed searching environment.

Client/Server Interaction using the Z39.50 Ranked Query

When a client submits a Z39.50 Ranked Query, it has the option to instruct the server to reformulate the query to better describe the user's information need. The server modifies the query based on its knowledge of the collections it is searching, the vocabularies native to those collections, general linguistics, & the most effective expansions of the query terms as related to the desired precision & comprehensiveness specified by the client. If the client has so requested, processing can stop here, & the reformulated query is shipped back to the client for further modification by client and user.

The session-oriented `state-ful' nature of the Z39.50 protocol facilitates the following types of client-server interactions using the Ranked Query:

- The ability to refine a query through a series of server reformulations & client modifications & (re)submissions;

- The user can use selected results from previous searches as relevance feedback:
  - Entire documents or portions of documents may be referenced by the client in subsequent queries.

Z39.50 Ranked Query increases Client Control over Query Processing

The Z39.50 Ranked Query gives the client more control over processing & evaluation of the query:

- The client is able to restrict the set of documents (collection) to be searched by including Boolean search restrictions such as date, author, subject, etc;

- The user/client may provide a number of 'hints' (suggestions) about the importance of particular query components, for instance: weighting of terms & operators, use of a variety of special ranking operators, & query reformulation options (e.g, term expansion, linguistic relationship) options;

- Many search systems support a 'tuning' mechanism to adjust the relative importance of precision vs. recall. The Z39.50 Ranked Query allows the user/client to adjust this weighting using a standardized weighting factor.

Z39.50 Ranked Query allows the Server to Return Postings Information

Because of the less predictable & deterministic nature of relevance based searching (as discussed above), a search server may perform query modifications or complex processing which is unrelated to what was specified in the user query. Although a client has more control over Z39.50 Ranked Query processing, the whys & wherefores of server query reformulation are still quite difficult for the user/client to understand.

Thus, an important feature of the Ranked Query is the ability for the server to return search result demographic meta-data (often referred to in the IR industry as `postings' data). The format & content of this data is also standardized within the definition of the Ranked Query, making it easier to interpret `postings' data from many different types of search systems.

Library Of Congress Position Paper

Z39.50 and Navigating Distributed Collections

Ray Denenberg  Library of Congress

Position Paper Prepared for the Distributed Indexing/Searching Workshop, May 28-29, 1996

Libraries and other institutions are creating collections, organized thematically (e.g. by subject, creator, historical period) with numerous, diverse objects, both digital and physical. These collections are often organized hierarchically and distributed across institutions and servers.

Significant resources may be invested in digitization and in the intellectual efforts of aggregation, organization, and description of the information in a collection. Yet to a user or client, a collection often appears to be simply an accumulation of undifferentiated data, because there is no agreed-upon semantics for navigating the collection, to locate and retrieve objects of interest. Coherent organizational structures must be imposed on the data, to provide a view that supports navigation.

A key obstacle to effective navigation is the inability to distinguish content from description. A primary goal of navigation is to locate and retrieve objects of interest; a vital step in that process is to locate relevant descriptive information. Thus it is useful to navigate among descriptive information as well as content, and consequently, to be able to distinguish content from description.

Various ad hoc descriptive formats have been developed, to describe collections as well as objects (e.g. finding aids, encoded archival descriptions, exhibition catalogs). At many institutions, so-called descriptive aids of various types have been created, at significant expense, and it is imperative that an application relying on descriptive information for collections and objects be able to exploit these available aids, rather than mandate the creation of new, redundant structures. Often, however, these descriptive aids do not have a well-defined structure and cannot be used alone for reliable navigation.

Another navigational problem is posed by the various and often complex relationships among and between objects and collections. For a given object, there may be other objects that are duplicates or variations (e.g. different resolutions) or which bear other relationships (e.g. thumbnails). For any given collection there may be superior, subordinate, related, and context collections. These relationships among objects and collections must be modelled coherently.

Different digital objects, even within a single collection, may be retrievable by different protocols (e.g. Z39.50, HTTP, FTP). It is therefore important that structures describing these objects, and how they may be accessed, be defined independent of any specific protocol.

All of these problems amplify when a collection is distributed across institutions and servers. In particular some normalization of query semantics is necessary for coherent navigation of collections. Clients must be able to formulate queries that will be interpreted consistently across servers.

Solution

Z39.50 provides rich tools for effective navigation of digital collections. A Z39.50 *profile*: Z39.50 Profile for Access to Digital Collections has been developed and implementations are in progress. The profile addresses the problems cited above; it models collections, objects, descriptive aids, and defines an enveloping structure for describing an object or collection.

The profile, though limited in scope, explicitly anticipates the development, of several *companion profiles*, addressing specific disciplines, for example, museum collections and objects. The Digital Collections profile, together with the companion profiles, have several objectives. Among these are: to allow a server to clearly designate what is content and what is description; to allow a client to navigate among descriptive information; to model relationships among and between collections and objects; and to provide semantics for queries to be interpreted consistently across servers, and data structures that provide semantics for navigation, ultimately to allow a user to locate and retrieve objects of interest.

Los Alamos National Laboratory Position Paper

URCs as a substrate for distributed searching

Ron Daniel Jr.
Advanced Computing Lab
MS B287
Los Alamos National Laboratory
Los Alamos, NM, USA 87545
rdaniel@lanl.gov

The increasing number of resources on the web makes centralized indices less and less satisfactory. Some form of distributed cataloging and indexing effort seems necessary. But exactly what form? What sort of cataloging information should be collected? Who will create the descriptions and how will they be managed over the lifetime of the resource and beyond? What protocols will be used to transfer the descriptions? How will queries be encoded and what query facilites will be provided? What sort of forward knowledge must be propagated to allow reasonable query forwarding?

These questions cannot be answered once and for all. We must have a system that can adapt to change, that can allow many different experimental solutions to co-exist, while preserving to the greatest degree possible the intellectual investment that has been made in describing network resources. The library community has already shown us the power of shared cataloging.

Uniform Resource Characteristics (URCs) were proposed by the IETF's Uniform Resource Identifier working group as a structure for containing information on networked resources. The rough documents for URC standards specify an abstract service that can have many different concrete realizations, and specifies how those different realizations can interoperate. The key ideas behind URCs are:

Allow for a variety of attribute sets, known as "URC subtypes".
: An attribute set that is appropriate for describing HTML pages is not likely to adequately describe scientific datasets. Any reasonable indexing system must allow different descriptive schema to be used, and must address namespace conflicts between the schemes.

Standardize the meaning of a *very few* elements.
: Having a variety of descriptive schemes means that systems will frequently encounter descriptions in unknown schemas. However, elements such as URL, URN, URC, and Content-type have a rigorous definition and are so pervasive that standardizing them will allow a great deal of useful work to be performed even when the whole of the descriptive scheme is not known.

Don't specify one syntax, instead specify a canonical representation that can be mapped into and out of a variety of syntaxes.
: Specifying one syntax is a recipie for disaster over the long haul, and leads to religious battles in the short term. PICS, IAFA, SGML, and MARC have adherents for reasons. An appropriate canonical representation should accomodate all of these.

Standardize the basic operations to manipulate the canonical representation, and let different query and transformation languages be developed to utilize those operations in novel ways.
: Services will want to compete on the simplicity or power of their search capabilities. A means for allowing that will also allow search capabilities to gracefully evolve.

Don't specify one protocol, instead specify how the canonical representation and the operations on it are encoded in particular protocols.

At the W3C Workshop on Distributed Indexing/Searching I would like to present a summary of the current state of the URC effort and give some examples of their use for distributed resource discovery.
For more information, see http://www.acl.lanl.gov/URC/

Last Modified: 14 May, 1996

Lycos, Inc. Position Paper

Lycos: Distributed Indexing

Draft: submitted to MIT Distributed Indexing/Searching Workshop
Not for citation

Position paper for Distributed Indexing/Searching Workshop at MIT

Charles Kollar, John Leavitt, Michael Mauldin
Lycos, Inc.
555 Grant St #350
Pittsburgh, PA 15219
kollar@lycos.com, jrrl@lycos.com, fuzzy@lycos.com
412-261-6660

ADEQUACY OF ROBOTS.TXT

Although Lycos has always supported the current robot exclusion standard, we note two deficiencies. First, the existing standard as written contains ambiguities about the exact syntax, leaving implementors to make their own choices. Second, the server-based model breaks down for sites with weak central support, ie: on-line service based home pages, company sites, and universities.

We will propose a draft rewriting of the current standard (reducing the ambiguity without adding new features) at the workshop.

We also suggest two extensions to the exclusion standard:

1. Use of the **meta** tag in HTML to provide file-specific robot access information for that file. That way a user who can only control his or her own directory can still exclude file from robot access.

2. The ability to exclude a file or directory from robot access using a file naming convention (e.g.: a prefix of "ns-" or "prv-" on a path element would stand for "no spidering").

SUPPORT FOR MULTI-ENGINE SEARCH

Support for search combining services such as MetaCrawler or SavvySearch is controversial to commercial search services such as Lycos. These services pay for the substantial costs of collecting and indexing information by placing advertisements on the output results, and to the extent that meta-searchers strip those ads from the results, this practice is theft.

The best way to keep these services from being denied access to searchers is to agree on a model for carrying ads along with the content, and to identify to the original server (in the user-agent field, for example) that the request is coming from a meta-searcher.

DUBLIN META-DATA

Lycos acknowledges the usefulness of standardizing field names for common meta-data elements such as **subject**, **title**, **author**, and so forth, and history shows that consensus at any useful level of detail is difficult or impossible, so we would recommend either wholesale acceptance of the Dublin Core, wholesale acceptance of a competing standard, or skipping this task.

One caveat from our experience with large free-form documents such as found on the web. If users are allowed to attach their own indexing terms to meta data, there will be some amount of gamesmanship by authors attempting to make their documents come our first in ranked retrievals. Our philosophy is that documents are their own best description, and the inclusion of "invisible" search terms invites this practice of "spam-dexing".

ACCEPTANCE OF SOIF AS A STANDARD

Lycos is still evaluating the usefulness of SOIF as a standard, and takes no position at this time.

INTRANET ISSUES

One issue that must be addresses when dealing with the Intranet is that of compartmentalized access. On the Internet, public usally means public, and all documents can be treated identically. Within a typicaly corporation, there are pockets of documents that are viewable only by certain individuals, and these policies must be respected by any corporate wide information resource.

Last updated 19-Apr-96 by fuzzy@cmu.edu

Microsoft Corp. Position Paper

Distributed Indexing/Searching Workshop
to be held at MIT, May 28-29, 1996
Sponsored by the World Wide Web Consortium

Wick Nichols
Nikhil Joshi

Microsoft is interested in working with the Internet community to create standard conventions, APIs and protocols for distributed indexing and distributed query. At present we'd like to focus on distributed indexing as we feel it is a more tractable problem.

Some issues that we feel are important to address at this conference are:

1. A crawler needs a way to find a list of documents that have changed since its last visit.

2. End users want rich query functionality using full text, sentence and paragraph proximity, tagging information, etc. Administrators want to minimize use of bandwidth and system resources. How can we balance these conflicting goals in designing an interchange format? Anything but full text will cause indexers to lose information. Do we need formatting decoration? To what extent? To what standard?

3. How do we represent embedded information in an interchange format. With what markings, if any, to indicate that it was embedded (or linked)?

4. What is the minimum property set for an interchange format? We believe there should be one and an HTML syntax should be defined for it. The Dublin metadata set is a promising start..

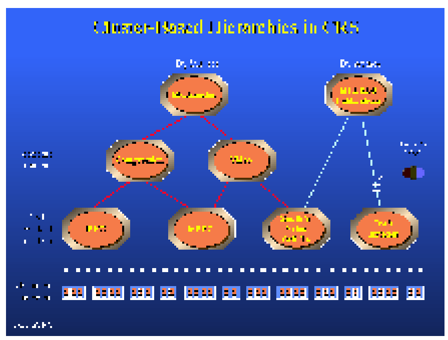
5. Is HTML or a related DTD a suitable interchange format? SGML is attractive because it is well-known and parsers are available. Such a solution can take advantage of other work such as the proposal to represent the Dublin Core in HTML.
Is there interest in a web-crawling consortium? This could take the form of a non-profit corporation, a for-profit corporation, an agreement to split the crawling problem between existing organizations. We are interested in discussing this.

MIT and Open Market, Inc. Position Paper

Services and Metadata Representation for Distributed Information Discovery

Mark A. Sheldon, Ron Weiss, Bienvenido Vélez, and David K. Gifford

Hierarchical organizations of information servers, such as *content routing systems*, provide a framework for distributed searching and browsing in large information spaces. Such hierarchies, as shown in the figure, consist of documents stored on document servers (e.g., Web sites) and information servers that organize and index documents and other information servers. Experience with several prototype content routing systems, including CRS-WAIS, Discover, and HyPursuit has led us to believe that effective search engines must provide a spectrum of browsing and searching capabilities, together with facilities for helping users focus queries. We have also learned that underlying metadata representations of server contents are critical to providing scalable implementations for high level information services.



Information servers must support interleaved searching and browsing activities ranging over a spectrum from a well-defined search for a specific document to a non-specific desire to understand what information is available. To support these activities, our systems use metadata information (*content labels*) to provide services that help *refine* user queries to focus a search, automatically *route* queries to relevant servers, and *cluster* related items.

Query refinement helps overcome the problem of excessively large result sets frequently returned by global searches by suggesting modifications to focus user queries. Relying exclusively on result set ranking functions is inadequate. Our prototypes dynamically compute and suggest terms that frequently co-occur with the user's query terms. When a query is sufficiently narrow to be efficiently processed, it may be automatically routed to relevant servers so the results can be merged and presented to the user. Metadata structures to support query refinement and routing include term collocation and frequency information.

The organization of information into clusters of related items assists both the users and the system in coping with large information spaces. The cluster abstraction allows a large information space to be treated as a unit, without regard for the details of its contents. A user exploring the portion of the information space relating to biology may want to identify all clusters (not all documents) that are related to DNA computation. Thus, the user may interact with the system at a level of granularity that is appropriate to the specificity of the information need and the complexity of the information space. Clusters also provide convenient units for the partitioning of work and resource allocation among the distributed components of the system. The HyPursuit prototype content labels incorporate metadata descriptions of clusters that consist of representative terms and documents.

We propose to investigate metadata representations that are extensible, scalable, and support the requirements outlined above. To insure that the architecture supports new user services, metadata representations should consist of an extensible set of information service specific components. To achieve scalability, the system must implement service-specific mechanisms to control information loss. The result of performing an operation, e.g., a search, on these content label components approximates the result of performing the operation on the entire information summarized by the content label.

NASA Position Paper

JPL Distributed Search Technology

David Wagner and Rick Borgen of the Jet Propulsion Laboratory have a strong interest in the Distributed Indexing/Searching Workshop. We have worked for 10 years plus on data management and archiving problems for the laboratory.

Our most direct experience is with an object description language (Keyword/Value Notation) developed by the Consulatative Committee for Space Data Systems (CCSDS), as used for distributed file management and long-term archive products. We also have recently developed a distributed search model and distributed search system known as the Distributed Object Manager (DOM), now in use by several laboratory projects.

Here are a few excerpts about our search system...

DOM is a general-purpose distributed cataloging system. It is general- purpose by means of a schema language that provides specification of types, common attributes, object attributes and collections. A client-server architecture with an SQL-like server protocol language supports flexible distribution. It is a catalog system in the sense that it maintains meta-data descriptions of well-identified objects and supports appropriate search and description features, but it does not try to support full traditional DBMS functionality.

DOM employs a kind of hierarchical organization of collections, except that multiple parent collections are possible. This structure, known as the collection lattice, is a principal organizing mechanism which supports attribute sharing, access permissions, search paths as well as the logical integration of multiple servers.

DOM also employs a type system for classifying objects, which can be considered another organizing feature that cuts across the collection lattice. The type system serves the traditional role of providing an attribute template for sets of objects. It also serves a very important role for supporting distributed queries based on object type.

The DOM system provides a systematic scheme for organizing large numbers of servers to work effectively together. The goal is to approach the kind of uniformity and simplicity in the distribution model of the World-Wide-Web, and yet support the kind of sophisticated queries associated with database systems. The collection lattice, the type system and common attributes are the essential mechanisms for accomplishing this multi-server integration.

Return to W3C Distributed Searching/Indexing

Netscape Commun. Corp. Position Paper

W3C Distributed Indexing Workshop: RDM/SOIF

Darren Hardy <dhardy@netscape.com>
Netscape Communications Corporation
May 6, 1996

As part of the Netscape Catalog Server project, Netscape has adopted and extended the Harvest distributed indexing technology via a mechanism called Resource Description Messages (RDM) which uses SOIF as its underlying syntax, and HTTP as its underlying transport protocol.

What is SOIF?

Harvest's Summary Object Interchange Format (SOIF) is a *syntax* for transmitting resource descriptions (RD) and other kinds of structured objects. Each RD is represented in SOIF as a list of attribute-value pairs (e.g., *Company* = *'Netscape'*). SOIF handles arbitrary textual and binary data as values, and with a simple extension handles multi-valued attributes. Also, SOIF is a streaming format which allows many RD's to be represented in a single, efficient stream.

What is RDM?

Resource Description Messages (RDM) is a mechanism to discover and access *Resource Descriptions* (RD) (or metadata) about network-accessible resources. RDM is implemented as a layer on top of HTTP, giving it the ability to leverage off of existing HTTP-based technology, and uses Harvest's SOIF technology to exchange indexing information over the network incrementally and efficiently. In addition, RDM supports a *Schema* which describes the SOIF, such as attribute names, data types, content types, and other information. RDM also supports a *Server Description* which describes some of the vital statistics about the RDM server, and provides a brief description the content of the server itself (i.e., with some sample RD's and a human-generated description). Finally, RDM supports a flexible scoping/view mechanism to access or search the RDs in a query-language independent fashion.

How is RDM/SOIF used?

RDM supports the Harvest Broker/Gatherer architecture. The Broker uses RDM to retrieve indexing information from a Gatherer; and an end-user search client uses RDM to send a query to a Broker and to retrieve the query's result set.

A *Gatherer* exports its Resource Descriptions (encoded in SOIF) via RDM to Brokers or other search engines interested in its indexing information. Typically, an automated Robot is co-located with the Gatherer to generate the indexing information for a collection of WWW servers.

Brokers or other search engines can use RDM to contact a Gatherer and incrementally download Resource Descriptions (encoded in SOIF). Also, if desired, Brokers can use RDM to download the schema or server descriptions from the Gatherer to customize their indexing algorithms.

NTT Corp. Position Paper

Ingrid is a research project at NTT Software Labs to design and build a suite of software tools that provide fully scalable, fully distributed web searching/discovery. The basic idea is that sites index their web documents locally, and are then automatically joined into a global distributed search infrastructure. A search on that infrastructure from any location will find all (or nearly all) matching web documents and related search engines.

At the workshop, I'd like to say what we're doing with Ingrid, give some information about the infrastructure we will have built at that point (hopefully it'll be something), and discuss what "open" protocols we use for Ingrid, including what extensions we need to the SOIF format (I'm assuming at this point that we will move to the SOIF syntax, though we have a number of new types.)

Ingrid uses two separate protocols. One is the Ingrid Access Protocol (IAP), which is a TCP-based thing (fairly similar to HTTP) used to submit items (in SOIF form) to the Ingrid infrastructure, and to search the Ingrid infrastructure. People can use this to build applications on top of Ingrid. The other is the Ingrid Control Protocol (ICP), which is a UDP-based thing used to create, maintain, and search the Ingrid infracture itself.

Online Computer Library Center Position Paper

Application for Participation:
Distributed Indexing/Searching Workshop

Stuart Weibel
Senior Research Scientist
OCLC Office of Research
http://purl.org/net/weibel

The Dublin Metadata Workshop of March 1995 and the Warwick Metadata Workshop of April 1996 were convened to promote the development of consensus concerning network resource description across a broad spectrum of stakeholders, including the computer science community, text markup experts, and librarians.

The result of the first workshop -- the Dublin Core Metadata Element Set -- represents a simple resource description record that has the potential to provide a foundation for electronic bibliographic description that may improve structured access to information on the Internet and promote interoperability among disparate description models. It's major significance, however, lies not so much in the precise character of the elements themselves, but rather in the consensus that was achieved across the many disciplines represented at the workshop.

The Warwick Metadata Workshop was a follow-on activity, intended to broaden the international scope of consensus and to identify impediments to deployment of a Dublin Core model for resource description. The results of this workshop include a proposed syntax for the Dublin Core, the development of guidelines for application of the Dublin Core, and a framework (the Warwick Framework) for metadata that will promote modular, separately accessible, maintainable, and encryptable packages of metadata. Thus, a Dublin Core package might be one of a number of other packages, including packages for terms and conditions, archiving and preservation, content ratings, and others.

As co-convener of these two workshops, I would be pleased to represent the results of these activities at the Distributed Indexing/Searching Workshop, especially with regard to the possible inclusion of the Dublin Core in HTML.

OpenText Corp. Position Paper #1

It is quite unlikely that the current paradigm of multiple Internet search facilities, each running a competing robot that is trying, on its own, to cover the web, will remain viable in the face of the continuing observed growth.

There are a variety of strategies with which this problem could be attacked. It is important to consider, for each strategy, its business implications as well as its technical viability. There is no point cooking up a dream technology that will seriously damage the interests of Netscape, Microsoft, and the flock of recently-IPO-funded Internet Index purveyors, because no such technology has any hope of adoption.

I propose an overview presentation that attempts to enumerate these strategies and briefly outlines the technology and business implications, pro and contra, of each. Some of the approaches that will be included in this tour are:

1. Dividing the crawling problem
 1.a Geographically
 1.b by subject domain
 1.c by network domain
This obviously has to be "Plan A". Since the problem is too big for everyone to solve on their own, basic computer science suggests partitioning it. Each partitioning brings with it a set of advantages and problems.

2. Unifying the robots Internet search facilities compete (presumably) on the basis of standard information retrieval metrics   such as precision and recall. But they all start at the same point: a lot of web pages. Is there a case to be made for centralizing crawling activity, and sharing its results?

3. Crawling less There is a lot of stuff on the web that needs to be crawled every day. There is a lot of other stuff that needs to be crawled only once [e.g. the text of the U.S. Declaration of Independence and of "Pride and Prejudice"], and there is a very large amount of stuff that arguably never needs to be crawled at all. Is there a basis for joint work in formalizing some metrics here?

4. Sharing the burden with the providers Being indexed is a net benefit to the providers of information. Given that current crawling strategies are likely to break down in the face of Internet growth, it is reasonable to ask them to participate with a small amount of the effort. Some ways they could do this are:
 4.1 Provider-push crawl requests
 4.2 Provider-generated metadata [subject category, volatility, etc.]
 4.3 Canonical URL notification - a huge win in terms of duplicate control

My personal position is that, to some degree, application of all of these strategies is essential for success. I am the chief designer and implementor of the Open Text Index and in particular its crawling/indexing subsystem, "Firewalker". - Tim

OpenText Corp. Position Paper #2

Position Paper: Internet/Intranet Indexing Suggestions

Open Text Corporation provides text search and retrieval tools for the Internet and for corporate intranets. We currently operate a free text search and retrieval service for a substantial portion of the published documents on the WWW. We also sell products that bring this functionality to companies wishing to provide similar facilities for their corporate intranets. Because of our experience at Open Text, a lot of suggestions have been made for improving the current standards (or defacto standards) with an eye towards improving our indexing and search technologies. The following list is a sample of some of the things that Open Text thinks would go a long way in improving the current search and retrieval experience.

Reducing Bandwidth Requirements

There's a lot of discussion currently on how to reduce the amount of data and tcp/ip connections required to obtain a page with all the embedded gifs, frame components, java applets, and other objects.

Ability to fetch multiple documents in one connection

A multiple get facility in the http protocol (similar to the mget facility in ftp) would assist greatly in reducing the overhead these connections entail. Robots can potentially utilize this multiple get facility to fetch several (perhaps unrelated) documents with one connection. The multiple gets should be robust enough so that unwanted data is not included in the transmission.

Server provided document lists

Robots periodically check to see if documents have changed on a server using the "if-modified-since" mechanism. While this saves transferring the document in those cases where the document has not changed, it still requires the overhead of one connection per document on the server to make this assessment. A better approach might be to provide a list of all documents available at the site, coupled with the size, last modification date, and mime type of those documents. Robots could collect this one file periodically, and from it infer which documents need to be re-fetched. On the server side, some form of daemon may be required to administer this file, but that should not be too difficult to create. This document could be given a reserved name similar to robots.txt, perhaps "sitedocs.txt". For large servers, this file could point to other files, perhaps reflecting the directory organization in use at the site.

Ability to "get-all-documents-modified-since"

An extension of the multiple file transfers and "if-modified-since" mechanism is to provide a "get-all-documents-modified-since" protocol to the HTTPD. With this, a browser or robot could make one connection and fetch all the documents from the site that have been modified or added since a specified time. Additionally the transfer could include a special block listing all URLs that were deleted from the site since the specified time.

Server side document conversion

Few servers currently provide mechanisms for converting documents on the server side. Converting large word processing or PDF files into smaller text or html documents before transmission could save a lot of bandwidth.

Document Change Frequency

Anything that can assist in defining the frequency at which documents are expected to change would assist in the currency of information available from search engines. While mechanisms like providing a mapping of URLs (perhaps defined by a regular expression) to expected change frequency, or expiry dates for documents, could assist, they could also be easily abused.

Improving the Robot Exclusion Protocol

The robot exclusion protocol provides some information on which robots are allowed to crawl what on a given site. A number of sites are excluding all robots because of perceived performance implications. There are a few things that could improve the situation so that robots minimize the impact of visiting and collecting documents from a site.

Time slots for robots.txt

The robots.txt file should detail time slots (in GMT) for crawling, perhaps on a per robot basis. These can be expressed with the same flexibility as the UNIX cron facility, so that sites can restrict robots to late night slots during weekends only, for example.

Bandwidth suggestions for robots.txt

The robot exclusion protocol provides guidelines on how often a robot can access a site for a document. These may not be appropriate given the wide variety of hardware and the popularity of some sites. Perhaps the robots.txt file itself should provide more site specific guidelines like how many files, bytes, or connections to make per unit of time, perhaps defined in time slots over the day, and perhaps by robot.

Refetching requirements of robots.txt

There should be some indication of how frequently the robots.txt file itself should be refetched. The administrators we contacted varied wildly on this one. The robots.txt file should contain an expiry date, or an indication of how frequently it should be re-fetched.

Improved Information Content

Identifiable summaries or abstracts

A short summary about the document is provided by most search engines. For now, those summaries are generated based on the content of the page at hand, which in some cases leads to difficult to read verbiage. A more formalized approach may yield better summary lists.

Improved keyword identification

Anything that can be done to improve the quality of search keywords in a document would assist search and retrieval engines. Our experience, however, has been that these mechanisms are generally abused on the Internet, but not on corporate intranets.

Improved character encoding and language identification

There should be a standard for character encoding(s) the identification of the character encoding scheme used. This is a particularly troublesome situation in countries like Japan where UNICODE, EUC and other encoding mechanisms are sometimes at odds with each other. In a similar vein, knowing what natural language(s) a document is in (ie English, French, Japanese, etc) would allow a search engine to tailor results for a particular user.

Site Specific Information

There are a number of improvements that can be made on a per site basis.

A graphics logo file per site

Each site should provide a small graphics file that represents the site, perhaps a corporate logo. These logos could be used to improve the appearance of a summary page, provide a more graphical means of navigating the net, or other unforeseen applications. This file should probably have a reserved name like robots.txt, perhaps "sitelogo.gif".

Geographic location of site

It should be possible to the determine geographic location of a server, either from the HTTPD or from an auxiliary file located on the server site, perhaps "location.txt". Robots can use this to optimize the collection of documents, and applications can be tailored for regional requirements.

Site summary

Sites could provide summary information describing the nature or purpose of the documents provided by the site, perhaps in a file called "summary.txt". For sites providing many services or types of documents, this file could allow a number of summaries organized by regular expressions defining the URLs that are associated with those summaries.

Multiple server sites

Many larger HTTP servers are implemented using several machines, and hence multiple IP addresses and hostnames. There are benefits in knowing both the IP address and hostname of the "main" server. The functionality already exists in the DNS protocol. However in order to implement this, a webmaster needs to know DNS setup intricately. As well, the webmaster must have the control of their sites' DNS. Many webmasters do not have this control as it may be considered a system administrator's or even the ISP's control.

Oracle Corp. Position Paper

Subject Indexing on the Web

David W. Robertson

Member of Technical Staff, ConText Server Group

Oracle Corporation

Current distributed indexing and search mechanisms often suffer from a lack of precision in results returned from a query. Searches are beginning to be based on the concepts as well as words present within a document. Searches on concepts increase the ratio of desired documents to irrelevant documents returned.

One way that precision has increased is through the use of subject catalogs. Typically, the main themes of a document are determined by a human cataloger, instead of automatically. Manually determining the subjects of a large percentage of documents on the Web is a daunting proposition.

The ConText option included in Oracle7 allows automatic classification of documents by subject. Indices reside in a standard Oracle database, providing security and fault tolerance. Automatic classification is made possible by natural-language processing using an extensive dictionary, and helps to solve the problems discussed above.

Another problem in obtaining relevant information is the stress that search engines place on Web servers and the network in building their databases. The Harvest System addresses efficiency concerns in Web crawling and searching through the concepts of Harvest Gatherers and Brokers.

ConText would be a useful commercially available index/search back/end for those using Harvest Brokers and for those using SOIF. Can Harvest be extended to provide adequate support for subject attributes, through SOIF and the Essence subsystem, for use with search engines such as ConText. For example, ConText can extract the major themes of a document. Provisions for attributes specifying multiple subject headings would be useful. Support for mapping subject headings from one classification system to another would also be helpful.

See Distributed Indexing/Searching Workshop for other papers prepared for the May 28/29, 1996, World Wide Web Consortium meeting in Cambridge, Massachusetts.

PICA - Centre for Lib. Automation Position Paper

Z39.50 and multi-national/multi-lingual environments

Makx Dekkers, Pica (Netherlands)
Position Paper Prepared for the Distributed Indexing/Searching Workshop, May 28-29, 1996

As a result of the possibilities offered by network technologies of the past decades, exchange of information via computer has become an every-day phenomenon in today's world. However there are problems in multi-national and multi-lingual environments that are not always obvious in the U.S. These problems are often related to cultural differences: national bodies are responsible for national rules for bibliographic descriptions; more fundamentally, different countries use different languages with different character sets and sorting rules. Other problems include: different rules for conversion from 8- to 7-bit ASCII for indexing dependent on country and language; difficulties translating system messages, variety of formats, and differences in cataloguing rules between countries.

An illustrative example of the latter problem is the treatment of multi-volume publications: under some rules these are catalogued as one single record with repeated elements, under other rules they are described as separate entities with relations between them. Another example is the use of standard phrases in national language within the cataloguing rules, such as for title changes for journals (in Dutch cataloguing the description would contain the phrase "Voortgezet als:"). International exchange of such bibliographic descriptions would ideally involve automatic translation; however, this is not done in practice.

For searching, a major problem is that standardised keyword lists are usually defined in national language and subject code systems are also agreed in a national context. All these different language and country related rules and practices cause incompatibilities that are difficult or sometimes impossible to overcome.

A number of problems are associated with differences in character sets. Many scripts are used in the world and most existing library systems are unable to handle them all. Transcription rules or character set conversions sometimes lose information as are not always reversible.

In sorting the situation is even more complicated. Where the same character set is used in two languages, sorting order might be different. In some languages "o-umlaut" is sorted as "oe", in others it might appear at the end of the alphabet. Even when the same language is used in two countries, there might be differences in sorting order of names: in Belgium a personal name of "Van Dam" will appear under "V", in the Netherlands under "D".

Solutions

Solving these problems requires the introduction of negotiations, on-the-fly conversions, and powerful explanation techniques. Negotiation aims at establishing a mutually agreed environment allowing exchanged data to be converted from one format to another. If that is impossible, the user should receive information to explain why the result is not as expected and to suggest alternative actions.

In the latest published version of the Z39.50 standard, Z39.50-1995, mechanisms are incorporated to negotiate the use of character sets as well as language. This is a big improvement compared to the 1992 version of the Z39.50 standard. Z39.50 now supports multi-lingual systems. Character sets that can be used are ISO 10646 and ISO 2022, or mutually agreed private character sets. A client/server pair may agree to the languages to be used for server message (including diagnostics) intended for display to a user.

For data formats, it is clear that national or local rules will prevail to determine how information is stored in databases. A positive development in some European projects is that implementors are trying to build table-driven, public domain toolkits, both for format conversions, as well as character set conversions. Although 100% accuracy in conversion cannot be achieved, this might help in broadening the scope of Z39.50 interoperability.
In areas where negotiation or conversions cannot solve the problems, the use of the Explain facilities defined in the Z39.50 standard provide the solution. This facility is probably the most powerful feature of Z39.50. Through Explain, the user is given information to understand better what goes on behind the scenes and to allow him to make sense of the results of certain actions. Fortunately, all messages in Explain have been designed for multi-lingual environments.

In conclusion, Z39.50 provides a very useful tool for information retrieval but it is clear that differences in language and culture have an impact on its scope and usefulness in international contexts. Internationalisation of the standard has solved some of the problems. Hopefully, through the implementation and use of Explain some of the others can be explained to users. The aim should be to make it possible to provide services to a wide international audience, respecting the multitude of cultures and languages in the world.

Polytechnic Univ. Position Paper

A Unified Model and a Search Framework for Spatial Metadata

Ashish Mehta, David Rubin
Center for Applied Large-scale Computing (CALC)
Polytechnic University, Brooklyn, NY 11201
Phone: (718) 260 3305, Fax: (718) 260 3930
E-mail: amehta@quasar.poly.edu

Currently, geo spatial metadata are stored using various standards. These standards include DIF (Directory Interchange Format), GILS (Government Information Locator Service), CSDGM (Content Standard for Digital Geo-spatial Metadata), etc. If a user wants to retrieve metadata which are stored in more than one standard, it is a very difficult task to retrieve such metadata. This is because: 1) data elements of one standard differ from data elements of other standards and vice versa; 2) the same data element of geo-spatial data is named differently in different standards. There is no tool available which can reliably convert metadata from one standard to another and return results to the user.

Using object modeling techniques we have developed a unified metadata model and a search framework for metadata stored in various standards. Our approach has the following advantages: 1) Current metadata standards are modeled using the entity-relationship modeling technique. We have applied object modeling techniques to various metadata standards; 2) The unified metadata model is a repository for all the geo-spatial data elements. Once metadata are stored using the unified metadata model, it will be complete and will provide universal metadata access. 3) Due to inheritance properties of the object model additional standards can be easily added to the unified metadata model. It is a very difficult task using the current metadata standards.

First of all we have combined all the existing data elements and concepts into a unified data model. This unified metadata model has access to a search framework which supports metadata conversion from one standard to another. The Search Framework has two components: Multimodel Schema, and Conversion Framework. The Multimodel Schema contains searchable fields (data elements/attributes) and equivalent fields which require conversion. These attributes are arranged in a tree hierarchy. Classes (of the Unified Metadata Model) which contain search attributes and equivalent attributes are added into the Conversion Framework. Each class inherits conversion properties from the built-in conversion classes.

Raytheon Company Position Paper

Extensible Domain-Specific Metadata Standards

Shirley Browne, Kay Hohn, and Tim Niesen
Reuse Library Interoperability Group

Position paper for WWW Consortium Distributed Indexing/Searching Workshop

The Call for Participation asks whether the Dublin metadata specification should be added to HTML. We argue that a universal metadata specification should not be standardized as part of HTML because such a standard will not be universally applicable. Instead, we propose that domain-specific standards bodies develop metadata standards for their domains and bind those to existing Web standards such as HTML, SGML, and Z39.50. As an example of how this can be done, we offer the Reuse Library Interoperability Group (RIG) work on the Basic Interoperability Data Model (BIDM) for the software reuse domain, and on Web bindings for the BIDM.

The BIDM, which is an IEEE standard, is a minimal set of metadata that a reuse library should provide about its reusable assets in order to interoperate with other reuse libraries. The BIDM is expressed in terms of an extended entity-relationship data model that defines classes for assets (the reusable entities), the individual elements making up assets (i.e., files), libraries that provide assets, and organizations that develop and manage libraries and assets. The model was derived from careful study and negotiation of the commonalities between existing academic, government, and commercial reuse libraries, by representatives from these libraries. Reuse libraries need not adopt the BIDM internally, although many have. They can continue to use internal search and classification mechanisms appropriate to their unique missions while using the BIDM as a uniform external interface. The current work on Web bindings aims to map the abstract data model to concrete syntax specifications that can be used for interchange of asset metadata via the World Wide Web. The Web bindings, one that maps the BIDM to an SGML Document Type Definition (DTD), and another that maps to META and LINK tags in the header of an HTML document, have been defined and are currently being implemented and tested. Several participants are using the Harvest Gatherer to collect and interpret the metadata, using the Gatherer's SGML processing capabilities, but other SGML tools may be used as well.

Of course one needs knowledge of the semantics of the data model to interpret and process the metadata appropriately, and this knowledge may be obtained by reading the BIDM document. However, it would be advantageous to be able to transmit this meta-model information as well, so that it could drive interpretation of the asset metadata automatically. Furthermore, one extension to the BIDM has already been defined (the Asset Certification Framework) and another is underway (the Intellectual Property Rights Framework). Individual libraries may have additional metadata, beyond that specified in the BIDM, that they would like to make available, and may wish to extend the BIDM for this purpose. Thus, work is underway on a formal meta-model for describing the basic model and extensions to it.

We hope that groups in other domains will benefit from our experiences in developing and implementing an extensible data model for the software reuse community. We believe that the extended entity-relationship data modeling technique is a powerful way of capturing and describing metadata about network-accessible resources. We also believe that the RIG has achieved the proper balance between domain-specific standardization and domain-independent standardization, by developing an abstract semantic domain-specific data model and mapping the abstract model to concrete domain-independent representations such as SGML and HTML.

Stanford Univ. Position Paper

Informal Internet Standards at Stanford

Prepared by Luis Gravano
(Joint work with Kevin Chang, Hector Garcia-Molina, and Andreas Paepcke)
Stanford University

Document databases are available everywhere, both within the internal networks of the organizations and on the Internet. The database contents are often "hidden" behind search interfaces. These interfaces vary from database to database. Also, the algorithms with which the associated search engines rank the documents in the query results are usually incompatible across databases. Even individual organizations use search engines from different vendors to index their internal document collections. These organizations could benefit from unified query interfaces to multiple search engines, for example, that would give users the illusion of a single big document database. Building such "metasearchers" is nowadays a hard task because different search engines are largely incompatible and do not allow for interoperability.

Given a query, a metasearcher has to perform (at least) three tasks to provide a unified interface over a (large) number of document databases:

- Choose the best databases to evaluate the query
- Evaluate the query at these databases
- Merge the query results from these databases

The existing search engines do not help with the three tasks above. In general, text search engines:

- Do not export information about the sources (the **metadata** problem)
- Use different query languages (the **query-language** problem)
- Rank documents in the query results using secret algorithms (the **rank-merging** problem)

To improve this situation, the Digital Library project at Stanford is coordinating among search-engine vendors (Fulcrum, Verity and WAIS) and other key players (Hewlett-Packard Laboratories, Infoseek, and Microsoft Network) to reach **informal** agreements for unifying basic interactions in these three areas . We have also received input from representatives of GILS, Harvest, Netscape, and PLS. In particular, our proposal specifies the summaries of the source contents that the search engines should export to assist in database selection (e.g., these summaries include the vocabulary of each source). We also define a simple, extensible query language with commonly supported features, drawing heavily from the Z39.50-1995 standard. Finally, we identify the information that the search engines should return with the query results in order to merge multiple document ranks meaningfully.
(Latest draft of the informal standards.)

This position paper is for the Distributed Indexing/Searching Workshop to be held at MIT on May, 1996.

Sun Microsystems Position Paper #1

Integrating Heterogeneous Search Engines
Position Paper for the
W3C Distributed Indexing/Searching Workshop

Gary Adams and W. A. Woods, Sun Microsystems Laboratories, Chelmsford, Mass.
contact: Gary.Adams@East.Sun.Com, William.Woods@East.Sun.Com

Introduction

Integrating heterogeneous search engines will require protocols for communicating with search engines about their capabilities and for reporting information in result lists about scoring method used and about what constitutes a *hit*. The growing diversity of search methods poses interesting challenges to integration that can be addressed if there are sufficiently expressive protocols. For example, the Conceptual Indexing System being developed at Sun Microsystems Laboratories, is a *concept matching* engine that reports a *penalty-based* score for *dynamically* identified text *passages*. In this *dynamic passage retrieval* system, scores are assigned to regions of text determined at query time, based on groupings of query terms or conceptually related terms. This differs from *document retrieval*, which generates scores for entire documents, and from *static passage retrieval*, which identifies rankable passages at indexing time. Integrating this system with a traditional system requires a way to identify dynamic passages and a way to know that smaller penalty scores are better.

Negotiating about Engine Capabilities and Reporting Results

A multi-engine search system may want to interrogate a search engine to determine its capabilities or to negotiate with the engine about what information it wants. For example it may want to determine if a given engine supports a proximity operator, and for those that do not, pass the results through a postprocessing filter. A system that integrates heterogeneous results may want to ask a search engine to report the following kinds of information for each returned hit, if available:

- what score was assigned by the engine
- what scoring method was used
- what query terms were matched
- what were the corresponding hit terms (which may be related, but different)
- what were the term frequencies, if known
- what were their positions, if known
- what is the size of the document
- what is the size of the collection (number of documents)
- what are the document frequencies of the terms in the collection, if known
- what are the word frequencies of the terms, if known
- what are the positions (ranges) of hits within the document

One could use SOIF notation to make such requests. For example, the following might be used to specify desired capabilities, and a similar format could be used to report available capabilities:

```
@CAPABILITIES-REQUEST {labboot:9112
POSITIONS{1}: Y
SCORES{1}:    Y
WORD-FREQUENCIES{1}:  Y
SCORE-TYPE{33}:       TWIDF,IDF,PROB,WORD-COUNT,PENALTY}
```

Returning a result list as a collection of SOIF objects would give a way to encode collateral information about results. For example, the following could be a passage retrieval result:

```
@DPASSAGE { http://www.sunlabs.com/
SCORE{3}:      .01
SCORE-TYPE{7}:        PENALTY
PASSAGE-REGION{11}:   01736,01895
HIGHLIGHT-REGIONS{23}:        01754,01799 01804,01815}
```

References

- Darren R. Hardy, Michael F. Schwartz, and Duane Wessels, Harvest User's Manual, U. Colorado, January 31, 1996.
- EARN Staff, Request For Comments 1580, "Guide to Network Resource Tools", EARN Association, March 1994.
- J. Foster, ed., Request For Comments 1689, "A Status Report on Networked Information Retrieval: Tools and Groups", University of Newcastle, August 1994.
- Conceptual Indexing Fiscal 1995 Project Portfolio Report, Sun Microsystems Laboratories, November 1995.
- Sun Microsystems Laboratories Knowledge Technology Group -- Conceptual Indexing Project home page, Sun Microsystems Laboratories, February, 1996.

Call for Participation

Sun Microsystems Position Paper #2

Distributed Indexing/Searching Workshop Position Paper

Wayne C. Gramlich

The Internet is ripe for some search and index standards. Currently, most searching and indexing technology tends to be rather monolithic. The Harvest architecture provides a perfectly adequate starting point for thinking about how to break search and index technology into smaller and more modular pieces. However, there are some additional places for some standardization above and beyond the Harvest SOIF interface:

TQL (Text Query Language)	The text search industry would benefit greatly from standardizing on a search language, just like the relational database industry standardized on SQL (more or less.) Users would benefit because they could learn and master one language instead of a multitude of similar but frustratingly different text query languages. The language needs to be designed so that the query features that are present in almost all query engines are readily available, while still preserving accessibility to higher level functions that are only implemented in one or two query engines. The design of TQL will be quite challenging and controversial, but ultimately will be quite well received by the user community.
Spider Helper	Right now, spiders have to continually reprobe the documents to ensure that they have not changed. This wastes network bandwidth and time. There needs to be interface that allows web spiders to find the documents that have changed since the last time they probed. This can be organized as a fairly simple CGI script that returns the list of all documents that have been modified/added/deleted since a specified time.
Distributed Query Support	Right now each query engine implements its own algorithm for rating query matches that is different from all other query engines. While a standardized algorithm is one possible solution, it is unlikely that there is one "best" algorithm. Instead, it possible to contemplate an interface directly to the inverted index that by-passes the query engine. This interface would provide the ability to query a document collection with a list of words and get back a list of document names and the word positions (for proximity search) in the documents. A centralized query engine can collect the information from a set of document collections and form a coherent match list using same rating algorithm rather than trying to merge a multitude of different query engine results. Again, this functionality can probably be shoe-horned into a CGI script to speed deployment.
Document Filter Standardization	Right now, each search engine vendor has to write their document filter code that extracts words from documents prior to insertion into the the query index. When an organization comes up with a new document format, the organization has to go around to all of the search vendors and ask them to write a filter for their document format. It would be so much easier if a standardized document filter interface could be defined that attached to the Harvest SOIF interface. This would greatly simplify the search vendor's lives as well as the new document format organization's life. Such a standard filter interface could easily be added to the Harvest SOIF interface.
HTML Meta Tags	Right now HTML only supports the definition of the <TITLE> tag in the <HEAD>. It would be relatively easy to expand this list to include important information such as language, authors, publisher, publication date, E-mail address, keywords and phrases, etc. This can be done using a combination of the <META> and <LINK> tags so that there is

no need to wait for browser vendors to implement new HTML tags. In addition, it makes sense to define a new HTML <ABSTRACT> tag to explicitly delineate the abstract portion of a paper if it exists. Similarly, it would be useful to define some tags to support bibliographic entries as well. Many search engines would be able to usefully use this additional document information.

While there are other opportunities for standardizing interfaces for search and index functionality, I believe that standardizing the interfaces above will be the most fruitful.

Wayne C. Gramlich

Sun Microsystems Position Paper #3

Call for Participation

Multilingual Issues in WWW Indexing and Searching

Position paper for the W3C Distributed Indexing/Searching Workshop

Philip Resnik and Gary Adams
Sun Microsystems Laboratories
philip.resnik@east.sun.com
gary.adams@east.sun.com

Introduction

The World Wide Web is an international phenomenon, yet its infrastructure is at present ill equipped to help users deal with languages other than those with which they are familiar. With the advent of Unicode, browsers that seamlessly support the display of multiple languages are not far off, but thus far little has been done to address the issue of multilingual *content*. As things stand, most of the popular Web search engines do have pages in multiple languages appearing in their indexes, but they provide no multilingual support to speak of, either at indexing time, at search time, or by way of helping the user cope when confronted with foreign-language text. This position paper is intended primarily to flag some of the issues that need to be addressed if standards for distributed Web searching and indexing are to take seriously the multilingual nature of the World Wide Web.

Indexing

Unless one adopts an IR framework based on character subsequences, indexing depends on the identification of meaningful units, typically word forms or word stems. Some key issues include the following:

- Identifying the language of the text
- Mixed-language documents: document-level vs. passage-level retrieval
- How to segment text into words (e.g. Japanese)
- Stemming and morphology (e.g. German compounds)
- Punctuation conventions

Query Processing

In addition to the same set of issues that arises at indexing time, processing user queries also raises the following questions:

- Character set issues when entering queries on forms
- Restoring accents to query terms that omit them
- Dealing with variant spellings

Conceptual Matching

"Conceptual" is something of a recent buzzword in the information retrieval business. Within a single-language setting, the general issue is locating text that might not use exactly the same words found in the query; for example, a search involving "agriculture" might do well to turn up documents about "farming". Multilingual retrieval is in a sense a generalization of this problem: a search for "computer science", viewing that term as a concept, should turn up instances of that concept even when expressed in another language, e.g. as "l'informatique".

Presentation Issues

If a search turns up hits in multiple languages, that still is not the end of the story: support must be provided for users who may not be familiar with all the languages they are faced with in response to a query.

- Identifying the language of the hit
- Optional filtering to exclude hits in unfamiliar languages
- Alternatively, help for "getting the gist" in unfamiliar languages
- Pointers to on-line solutions for translation

At Sun Labs, we have been working on a pilot project designed to help users "get the gist" of pages in unfamiliar languages, in order to decide whether to avail themselves of on-line opportunities for getting documents translated.

References

- Andrew Pollack, "A Cyberspace Front in a Multicultural War", New York Times, August 7, 1995, page D1.
- F. Yergeau, G. Nicol, G. Adams, and M. Duerst, "Internationalization of the Hypertext Markup Language", Internet draft draft-i etf-html-i18n-03.txt, February 13, 1996.
- T. Berners-Lee, and D. Connolly, "Hypertext Markup Language - 2.0", Request for Comments 1866, MIT/W3C, November, 1995.
- Co nceptual Indexing Fiscal 1995 Project Portfolio Report, November 1995.
- Sun Microsystems Laboratories Knowledge Technology Group -- Conceptual Indexing Project home page .

Tele2/SwipNet Position Paper

The Case for SOIF-2

Jeff Allen
<jeff@bunyip.com>
Bunyip Information Systems, Inc.

Patrik Fältström
<paf@swip.net>
Tele2 AB

May 1, 1996

Many wide-ranging application level protocols now under development in the networking community have a need for a compact, easy to manage encoding of a set of key/value pairs. This includes groups working in all of these areas:

- Indexing (Harvest's SOIF format, Whois++ templates, IAFA templates)
- Personal Data Interchange (Versit's PDI specification, the new application/directory MIME format)
- public key infrastructures (PGP, SPKI, PKIX and other X.509 profilers)

There is a clear need for progress towards a standard text representation with enough flexibility to handle these applications and more. Interoperability and implementation will be easier if the community can work towards a more standard template format. The first SOIF specification is a good starting point, but there is a good case to be made for SOIF-2.

Based on our experience, first with IAFA templates and more recently with the development of the Whois++ technology, we have distilled a list of issues designers of attribute/value encoding schemes need to consider.

Hard/soft newlines The template representation must be able to distinguish between newlines existing in the attribute value and those inserted in order to facilitate transport through systems with line length limitations.

Complex structures Some applications require either the ability to have attribute values that are themselves attribute/value pairs, or have some mechanisms for definition of relationships between attributes. In either implementation, a further consideration to be addressed is how (or whether) to specify individual subcomponents.

Template representation character set In order to distinguish between attributes and values, a clear definition of the characters used in the template representation must be available (e.g., delimiters, ans the associated quoting rules).

Attribute value character set The template representation must be capable of handling the character set(s) necessary to store data in a variety of languages.

Attribute metainformation Some applications require that metainformation regarding the language, format, or other qualities of the value be carried along with each attribute/value pair.

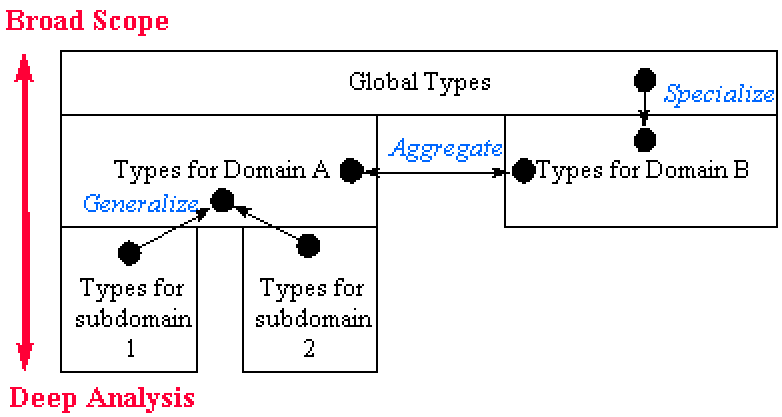
SOIF, as it is currently defined, carries the attribute name, the attribute length (in bytes), and the attribute value. Handling of line breaks and binary attribute values is incompletely specified. Metainformation and character set issues are not addressed at all. Some of the issues outlined above are handled more completely in other template systems, (line breaks in Whois++, metainformation in Versit).

The SOIF specification is halfway to a solution to a problem shared by many applications. The question is, "should we try to join these applications by a common template format"? We believe such a standard would be a very important step forwards towards creating interoperating indexing services. SOIF-2, an extended form of the current SOIF templates, should be specified and deployed as quickly as possible. It should address the issues above in simple ways, making the SOIF-2 specification useful in a broad variety of applications.

Transarc Corp. Position Paper

Position Paper for DISW'96

Mic Bowman



While the value of structure--i.e. discrimination of (meta)data by field--to improve the precision of query resolution is undeniable, it is rarely used effectively in the centralized Web *super-indexes* like Altavista and Lycos.

The current Web architecture requires a tradeoff between depth of analysis and breadth of scope. To be used effectively, structural definitions must span the collected resources. Within the scope of a single domain--either topical or administrative--an index can specify and enforce a standard schema for data with rich syntactic and semantic meaning. In the global web, however, autonomy vastly limits the acceptance of any standard. A centralized index is forced to use a "least common representation" for any structure that exists. In practice this means plain text.

We propose a framework for representing shared structure through a hierarchical type system. At the top level, very general document types specify a restricted set of structural elements; e.g. the fields specified in the Dublin core. Restricted domains such as the HPCC Software Reuse Interoperability Groupt (RIG) define domain-specific subtypes of the very general global types. For example, the RIG has a metadata standard called the Basic Interoperability Data Model (BIDM). Subtypes like those in the BIDM inherit the structural definition (i.e. the schema) of their parents and add additional fields. At the lowest levels of the hierarchy, individual users add personal elements to the structured data that is collected.
This approach has several benefits:

1. *It enables a full spectrum of search from deep analysis to broad scope.* A search begins with the most general types; i.e. those high in the hierarchy. For refinement the results of the search can be restricted to a particular domain to increase the expressiveness of the query.
2. *It can be implemented and deployed.* The short-term requirement for implementation is a syntax for describing a type. Simple extensions to SOIF like those proposed by Netscape for their catalog server, can already achieve this. For a high quality service, tools for type validation and evolution are required.

3. *It can be extended without need for global agreement.* Given a standard representation, any organization can define and enforce a collection of types. To share less detailed representations globally, the organization should choose types that are derived from the most general, global types.
4. *It enables cross-domain access through customized translation operations.* When necessary, deep analysis of independent type hierarchies is possible through the use of translation functions. This technique is commonly used by the designers of federated databases. The type hierarchy is used for most shared access since the necessary translation functions are expensive to design and implement.

My intent for this workshop is to begin the definition of standards that increase the availability and usefulness of structured data. I believe that a common representation for structured schemas like the hierarchical system we prpose is possible and would be highly advantageous.

This page created by Mic Bowman (mic+@transarc.com)

Last modified: Fri May 17 13:31:11 EDT 1996

UC Davis Position Paper #1

Position Paper for Distributed Indexing/Searching Workshop: Client Query Proxies

Chris Weider
Ken Weiss

Problem Statement

Distributed searching tools have proliferated in the absence of any standards for query syntax or resource discovery. As a result users must become familiar with both the contents and query rules for interacting with a variety of search engines. While there is some commonality provided by the general adoption of HTTP/HTML for managing the interaction with the user, this does not extend to the user interface itself.

URNs provide some of the framework for handling the resource identification aspects of this problem, but no architecture now exists to implement URNs across a broad spectrum of networked servers. A few projects have attempted to address the problem of mapping a standard query syntax into multiple information servers, but generally only within a single protocol (for example, Willow and Z39.50). There is some work going on at Berkeley as part of the digital library initiative there to generically map queries into servers, but the work is still in its infancy.

Proposed Solution

An architecture incorporating client query proxies can address many of the problems inherent in a distributed network of search engines. The query proxy can use HTTP/HTML to communicate with the user. Each user registers with the proxy server and provides information on his/her query syntax preferences. The preferences are stored on the query proxy, and discarded after a preset period of inactivity. When the user issues a query, the syntax is based on that user's preferences. The query submission also includes the server to search. The proxy then launches a process that contacts the server, downloads (and optionally caches locally) that server's query syntax, performs mapping from the user's syntax to the server's syntax, and issues the query. Query results are returned to the proxy, and passed back to the user. A block diagram of the architecture is available separately.

This approach requires the definition of a protocol for proxy-to-server communication, and standards for the definition of preferred query syntax. Ideally this would be handled similarly to Whois++, where the server can be queried for its templates and help files. However, in the short term it should be possible to agree on a port that will dump the necessary configuration information in a predefined format in response to a telnet connection from the query proxy.

To ease the problem of hand-coding the client query proxy with the query capabilities of the target server, it should be quite possible to add a function to search servers to describe, in a common vocabulary, the search and query capabilities of the server, to make the collection of this information feasible for machines. This dovetails nicely with the ideas described in our second position paper on **indexing proxies**.

Advantages of Proposed Solution

- Single client presentation can handle multiple query syntaxes - backend translation
- Server behavior can be customized to client - bandwidth, display, locale, budget
- User preferences are network-resident, accessible from anywhere

Disadvantages/Problems

- Few index servers now support passing templates & syntax metadata to client processes
- User still needs to know where to search

- User still needs to know what is being searched (WAIS, Z39.50, Harvest, Whois++...) so that proxy process can perform appropriate syntax translations

UC Davis Position Paper #2

Position Paper for Distributed Indexing/Searching Workshop: Indexing Proxies

Chris Weider
Ken Weiss

Problem Statement

Indexing systems such as Lycos and Alta Vista must actively download each and every document they wish to index. As services like these proliferate, and as the volume of information on the Internet gets larger, this will become increasingly more difficult to do in a timely fashion, or indeed to do at all. In addition, many of these services do exactly the same types of indexing on the documents. There may very well also be cases (particularly in private or semi-private networks) where it may be perfectly reasonable to export precomputed indexes while securing the documents themselves. This will become more important as the Net evolves to a) contain many more services which cost money, and are typically **not** indexed today, and b) many more access-controlled resources.

Proposed Solution

A system of indexing proxies should be developed and deployed which generate indices for the information contained in a given group of resources, and export them to indexing services. In this model, a system such as Lycos would contact the index proxy for a given resource site, ask for an index in a specific format, and then integrate the index into the rest of the service. This would also help prevent undesired replication of the entire data resource, a problem which is likely to become more prevalent as time goes on, particularly for smaller resources. It would also allow the integration of expensive resources into the search tools without requiring a substantial up front cost. This reduces the barriers to entry for many smaller special purpose index services.

In addition, if the indexing proxies set up indexing relationships with a number of services, the proxy can *push* any changed data without having to be polled for it. WHOIS++ successfully uses that model today.

Most types of indexing data can be propagated and perhaps even integrated together. Centroids, glimpse full-text indexes, WAIS indexes and so forth are all good candidates for transmission. This may have a disadvantage in that this may foster a reliance on a few types of indices, but this can probably be avoided.

This approach requires the definition of protocols and vocabularies for describing and transmitting indices, and perhaps standards for the specification of subsets of a given resource site.

Advantages of Proposed Solution

- Indexes can be created without access to the primary materials
- Commonly used indexes (inverted index, for example) can be cached and simply sent off
- Allows resources to be found with less risk of data dumping of the entire resource
- Allows services which provide highly dynamic pages (database access sites, for example) to participate in the commonly used index services, and describe their contents more fully
- Allows organizations to firewall access to resources while providing location information
- Potentially allows much more rapid update

Disadvantages of Proposed Solution

- New types of indexing will have to be built into the proxies over time
- May foster use of a small number of types of indices

UC Office of the President Position Paper

Position Paper Distributed Indexing/Searching Workshop Clifford Lynch University of California, Office of the President clifford.lynch@ucop.edu While the architectural model developed by the Harvest system provides a very valuable context for developing interface standards between web-crawlers and site-based gatherers, other recent work including the Warwick framework that is under development as a result of the recent OCLC/NCSA Metadata workshop and the work on the CNI White Paper on Networked Information Discovery and Retrieval suggests that this architectural model needs significant extension. Key areas include:

1. Modeling of site-based processes (algorithmic and/or intellectual) that might move from author-provided in-document metadata to more comprehensive external metadata "containers" that both enrich and refine the author-provided descriptive information and also may include (possibly through inheritance models) site policies regarding usage rights and other properties of objects. These external metadata containers might then be collected by gatherers for export or directly by web-crawlers. This has implications for the HTML extension standards that might be used to include or attach metadata to objects, as well as to the site (repository) interfaces.

2. The extension of descriptive information that is exported from a site (including what is part of the base SOIF element set) along the lines indicated by the Dublin Core work and including the interoperability insights obtained at the Warwick meeting. This would include, for example, the use of controlled descriptive vocabularies.

3. There is a need to improve support for collections of information at sites that cannot be directly indexed (for example, Z39.50 databases which are accompanied by EXPLAIN-based metadata) or which are not accessible to web-crawlers (or perhaps even to indexing algorithms embedded in Gatherers) except under highly controlled circumstances because the information provider wishes to retain control over what is exported (commercial intellectual property, for example). There is a need for a concept of trusted rendezvous sites where local information can interact with external indexing and abstracting algorithms, but where the information owners can be assured of some controls over the amount of extracted information that is being exported. The definition of these controls represents a key research problem.

UC San Francisco Position Paper

A Unified Element Vocabulary for Metadata

John A. Kunze, jak@ckm.ucsf.edu
Center for Knowledge Management
University of California, San Francisco

15 May 1996

It is generally agreed that a metadata record for an information resource contains descriptive elements that are suitable for use by automatic indexing programs (such as Web crawlers) and by citation display software (such as increasingly metadata-aware Web browsers). Each element has a name, whether or not a particular record syntax identifies it explicitly (e.g., "Author" in Author=Plato) or implicitly (e.g., the second unnamed element might by convention contain a "Title").

Problems arise in designing element name spaces (vocabularies) that fit current and projected metadata needs across many uses and fields of knowledge. One need is to support an ongoing process of enrolling new names into element vocabularies (extending them) while minimizing conflict with natural language connotations and existing element definitions. An appealingly simple approach is to partition a top-level name space into subspaces (that are perhaps further sub-partioned) for which enrollment and conflict management is distributed to various interested communities. While thus delegating name space management has clear strengths, some pitfalls bear highlighting.

One problem with partitioned element vocabularies is that the divisions tend to relax over time. For example, because boundaries between intellectual domains such as social sciences, humanities, and biology blur, communities pick up elements from each other and introduce inter-community dependencies and inconsistencies. As fields of knowledge advance at different rates, outdated divisions become more onerous and community interoperability drops. Moreover, experience with Z39.50 shows steady inclination in communities not just to pick and choose elements from each other, but to import external vocabularies whole into their own vocabularies. Another problem is that while hierarchical naming systems implied by partitioned vocabularies work well in software environments, they fit poorly in written and spoken communication, where they will be used often.

A Unified Element Vocabulary

A design worth exploring is a single vocabulary for all elements. Different definitions for the same element would appear together as alternates, just as in a natural language dictionary. Alternates would be tagged with the domain of origin (e.g., biomed). Such a list of elements could scale if need be a comprehensive dictionary containing one element per word of a natural language. To test whether an element vocabulary needs that much room to grow, it is sufficient to reflect how hard it is to think of a natural language word that could *not* conceivably name an element.

This vocabulary would consist of a stable base of approved elements augmented by an informally evolving set of commonly used elements. The vocabulary approval process would be lightweight and adaptable; an interesting functional model to borrow from is natural language, which has approved vocabularies (e.g., the Oxford English dictionary) augmented by a set of commonly used terms. An element vocabulary might also be amenable to categorization using concepts from the Warwick Framework, which partitions elements along functional lines.

Univ. Arizona Position Paper

Combining Browsing and Searching

(A position paper for the W3 Distributed Indexing/Searching Workshop, MIT, May 28-29, 1996)

Browsing and searching are the two main paradigms for finding information on line. The search paradigm has a long history; search facilities of different kinds are available in all computing environments. The browsing paradigm is newer and less ubiquitous, but it is gaining enormous (and unexpected) popularity through the World-Wide Web. Both paradigms have their limitations. Search is sometimes hard for users who do not know how to form the search query so that it is limited to relevant information. Search is also often seen by users as an intimidating ``black box" whose content is hidden and whose actions are mysterious. Browsing can make the content come alive, and it is therefore more satisfying to users who get positive reinforcement as they proceed. However, browsing is slow, very time-consuming, and users tend to get disoriented and lose their train of thoughts and their original goals.

We argue that by combining browsing and searching, users will be given a much more powerful tool to find their way. We envision a system where both paradigms will be offered *all the time*. You will be able to browse freely -- the usual hypertext model -- and you will also be able to search from any point. The search will cover only material *related* in some way to the current document. (Of course, global search may also be offered.)

As a first attempt to test this notion, we implemented a system that automatically modifies existing WWW pages to add search facilities such that the search domain from a given page depends on that page. Our system, called WebGlimpse, is based on our Glimpse and GlimpseHTTP search facilities, but it adds the concept of a *neighborhood* for each page. The neighborhoods are computed (based on options selected by the information provider) at indexing time (e.g., once a night). Only one index is used per site, but the index supports efficient search by neighborhoods. A typical neighborhood may be the list of all pages linked from a given page, all pages within distance 2, all pages in the same site, all pages pointing to that page, etc. WebGlimpse copies remote pages as well and automatcally adds them to the index. More complex definitions of neighborhoods, which may depend on semantic analysis, can also be added.

In summary, WebGlimpse allows any web site to offer a combination of browsing and searching by automatically analyzing the site, computing neighborhoods, and attaching search interfaces to existing pages. The search is efficient both in terms of time (neighborhoods are explored only at indexing time) and space (only one small index per site).

WebGlimpse Team: Udi Manber, Burra Gopal, and Michael Smith,
Dept. of Computer Science, University of Arizona.

Univ. Houston - Clear Lake Position Paper

DI/SW Position: Search and Meta-Search on a Diverse Web

David Eichmann

Research Institute for Computing and Information Systems
University of Houston - Clear Lake
Houston, TX

Introduction

One of the most significant challenges facing builders of indexing and search systems for the Web is the diversity of goals and capabilities of the content providers - the operators of the thousands of servers we so readily view as a single information resource. In operating the RBSE Spider [1], we've encountered reactions ranging from 'stay off my server' (usually expressed as a blanket exclusion clause in /robots.txt) to 'why haven't you indexed us yet?' (usually expressed in a mail message directly to me...).

A Tale of Three Prototypes

Two things remains clear through all of this - users want to track information relevant to their interests and are increasingly demanding efficient access to information. We are currently involved in the design, development and evaluation of three complementary systems to address these issues. The MORE repository system [3] is a meta-data based cataloging environment, providing separate hierarchies of meta-classes and collections and support for controlled access to proprietary collections through the definition of user groups. The RBSE Spider [1] retains both the structure of the Web in a relatonal graph representation and a full text index of the HTML documents encountered. The spider selects candidates for retrieval and indexing using a set of cached heuristics. The architecture readily supports multiple discovery modes through respecification of the candidate retrieval query. Sulla [4] is a user agent with the ability to acquire and act upon an interest profile of its user and the ability to act ethically [2].

The Pragmatics of Indexing and Search on the Web

Given the size of the Web and the diversity of its contents, how do you build a useful index? We've taken a non-traditional tack with the Spider. Our current architecture supports the exclusion standard, but also allows the operator to specify constraint patterns that candidate URLs must match against to be indexed and concept profiles (currently high relevance terms) that are used to rank newly identified URLs for indexing. The result is an index, that with only 40,000 documents, performs as well as Alta Vista in certain concept areas (e.g., agents and ontologies). Sulla interrogates a variety of search engines, each with its own search algorithm and scoring scheme. We've experimented with a number of approaches to merging returned results and have settle for the moment on the relative rank of a hit from a given engine as the basis for generating aggregate scores.

What Next?

The robots.txt file contains little information regarding server performance/load - and the rate its operator is willing to be accessed by an agent. On a global basis, this is our prime interest. On a local basis, we're shifting our tools from simple word indexing to concept indexing. A shared project in this area offers far more probability of success than convincing the world to do their own tagging.

Bibliography

1 Eichmann, D. ``The RBSE Spider - Balancing Effective Search Against Web Load,'' *First International Conference on the World Wide Web,* Geneva, Switzerland, May 25-27, 1994, pages 113-120.

2 Eichmann, D., ``Ethical Web Agents,'' *Proc. Second International World-Wide Web Conference: Mosaic and the Web,* Chicago, IL, October 17-20, 1994, pages 3-13.

3 Eichmann, D., T. McGregor and D. Danley, ``Integrating Structured Databases Into the Web: The MORE System,'' *Computer Networks and ISDN Systems,* v. 24, n. 2, 1994.

4 Eichmann, D. and J. Wu, ``Sulla - A User Agent for the Web,'' poster, *Fifth International Conference on the World Wide Web,* Paris, France, May 6-10, 1996, poster proc. pages 1-9.

Acknowledgements

This work is supported in part by a grant from Texas Instruments, Inc., by NASA as part of the Repository Based Software Engineering program, Cooperative Agreement NCC-9-30, research activity RB-02A, and by a grant from the Texas Advanced Technology Program.

Univ. Illinois at Urbana-Champaign Position Paper

Centralized and Distributed Searching

by Daniel LaLiberte, NCSA

There continues to be commercial value in attempting to index everything on the web, but estimates are that only a modest percentage of the actual documents have been indexed, while growth continues at an exponential rate. Meanwhile, users around the world search in these centralized indexes, or replicas of them, and the better search servers quickly become overloaded, thus reducing their effectiveness. Furthermore, the indexing that is done is very generic, i.e., not at all specific to the domain areas of the documents, thus reducing their usefulness. All these market pressures dictate that search servers will eventually need to specialize in various ways to hold on to an increasingly narrow niche. Automatic searching through a web of interlinked, related, domain specific indexes may be done many different ways. First, I describe the basic design of a centrally controlled search process, and then consider a distributed search process. Both types of search processes should be considered by those creating internet standards in support of searching; extensions to HTTP are briefly suggested.

Centrally controlled searching starts by evaluating a query at some set of nodes and continues on neighboring nodes. How the neighboring nodes are selected determines the overall behavior of the search. Neither a depth-first nor breadth-first search is generally appropriate because both effectively search blindly. Rather, we should continue to search only the parts of the web that have some promise of satisfying the query, so a measure of relevance for each match is required. Given a set of nodes with varying degrees of relevance, the neighbors of the most relevant nodes should be searched next. Furthermore, each link to a neighboring node should specify the relationship to the node and include a description of it; this info may be used in the relevance measure.

The basic query capability could be supported in HTTP with a SEARCH request given a URI for a collection to be searched (or '*' for the whole site), and parameters of the query in header lines. The PEP Protocol header would specify the type of query being used. The result of the request would usually be metadata for the resulting collection, or perhaps a small collection could be returned in a multi-part result. Metadata about collections and elements of collections may be fetched (for example, to be incorporated into a searchable index) by using a META request. A META request specifies the URI of the collection or elements and header lines to indicate which type of metadata is desired.

Centralized control of searching *seems* to be required to select the nodes to be searched, perhaps concurrently, and to merge the results, but now we consider how searching can be done with no central control. By distributing control, we gain much greater processing power with each remote server performing part of the search, and we avoid the bottleneck of a centralized controller managing the entire process. The results must ultimately come back to the originator, but even this merging process can be decentralized.

A distributed searching algorithm must overcome several problems. To avoid looping, queries MUST be identified and each node MUST remember the queries it has recently proccessed. Also, a query MUST include both a time-to-live and a maximum distance to limit its range in time and space.

Results from matching a query at a node could be sent directly back to the originator; this requires that the originator be available as part of the query, but moreover, the originator may be overloaded by too many incoming results. Better would be to merge the results along the reverse path of the query. Or perhaps nodes at local maximums (that match the query well) could be given responsibility to merge results of their neighborhood, which are returned to ancestor local maximums and ultimately to the originator. This last requires asynchronous notification of each merge node, which should wait up until the time-to-live before returning the merged results.

A problem for both centralized and distributed searching is that a query must be transformed into one that the receiving node understands, and similarly the results must be transformed on the way back out. This assumes heterogeneity in the search protocols available across the web, which I believe will always be the case because searching is the kind of problem that can never be sufficiently standardized for all search domains without going to the full generality of arbitrary function calls.

Univ. Queensland Position Paper #1

Resource Discovery and the Open Information Locator Project

Andrew Wood, Nigel Ward, Hoylen Sue, Renato Iannella

Research Data Network CRC, email: [woody, nigel, hoylen, ren]@dstc.edu.au

Resource Discovery is the term commonly used to refer to the exercise of locating, accessing, retrieving, and managing relevant resources from widely distributed heterogeneous networks. The Resource Discovery Unit of the Research Data Network Cooperative Research Centre is actively working on tools and technologies which make these tasks easier in the Open Information Locator (OIL) project.

The OIL project takes a broad approach to solving the resource discovery problem. We have constructed a definition of resource discovery and are developing a conceptual model as a foundation for building and evaluating our work. Included in our definition are the following assumptions about resource discovery: resource discovery is a global problem; resource discovery systems must be scalable; resource discovery is, in its very nature, distributed; resource discovery does not imply any fixed structure or hierarchy of information.

A prototype resource discovery tool called HotOIL is currently under implementation. The HotOIL system can be viewed as a match-making service between a user's query and a vast range of information sources. It is a testbed for our research into a number of resource discovery issues including **scalability**, **query routing**, **naming and meta-data**, **dynamic database access**, **information retrieval**, and **human computer interactions (HCI)** . This paper outlines our research into each of these issues.

Scalability. Our research focuses on the scalability issues associated with finding and accessing a large and growing number of information providers. Similar scalability issues are being researched by the open distributed processing (ODP) community. We are investigating the ODP trader and interworking technologies as a solution to this problem, and implementing a resource discovery trader based on the X.500 directory service. Information providers register details of their service as a service offer within such a trader. Federation of these interworking traders provides a solution to scalability.

The federated network of resource discovery traders is fundamental to our solution for the problem of **query routing**. The trader will be used to determine what information sources to query by returning service offers from information sources that are relevant to a query. This solution also has meta-data implications as it is necessary to provide a characterisation of the information providers.

The effectiveness of resource discovery systems will rely on flexible and extensible **naming** and **meta-data** mechanisms as a key to accessing resources. Most meta-data work, including development of the Dublin Core meta-data set, is concerned with describing documents. However meta-data describing large-scale queriable collections of resources is important. The HotOIL system requires meta-data about the type of information served by an information provider, and information about how to access that data. Our work in this area includes the successful implementation of a URN resolution service which resolves IETF proposed URNs into URCs described by an enhanced Dublin Core meta-data set, and the development of meta-data for queriable resources.

Dynamically accessible databases, are important to support the needs of global resource discovery systems. We are applying the Z39.50 protocol to this problem. The meta-data supported by the Explain database in Z39.50 offers a resource discovery system the ability to dynamically discover not only information about the interface to an information provider, but also information about the kind of information served by the provider.

Information retrieval techniques are important to address many of the issues facing the resource discovery community. In our work, attention has been paid to improving precision in the query result. We have implemented a Query By Navigation (QBN) system that uses nonmonotonic inference. We plan to investigate QBN in the framework of a `global' hyperindex. This hyperindex will be automatically constructed from the meta-data of distributed resources. The resulting hyperindex can be considered as a conceptual schema of the contents of these resources.

While much attention is being paid to the more mechanical aspects of resource discovery, we view the **HCI** aspects of a resource discovery tool also to be vital to its success. Aspects of the formulation of a users information need is being investigated as part of the QBN system. We are investigating resource visualisation techniques for the presentation and control of large result sets.

The first prototype of the HotOIL system is due to be completed in June 1996. It will integrate work from all of the areas mentioned above, and provide a platform to support our continued research in the field of resource discovery. Call for Papers

Univ. Queensland Position Paper #2

Z39.50/SQL+ - Stateful Web Access to Relational Databases

Robert M. Colomb and Sonya M. Finnigan

Distributed Database Unit, CRC for DSTC, University of Queensland, Qld 4072, Australia (email: s.finnigan@dstc.edu.au)

Distributed Indexing/Searching Workshop,MIT,May 1996,

http://www.w3.org/pub/WWW/Search/960528/cfp.html

Abstract: The ANSI/NISO Z39.50 Standard defines a protocol to facilitate the interconnection of computer systems for the search and retrieval of information in databases. In this paper, we present Z39.50/SQL+, the adaptation of this protocol to the SQL domain, and briefly discuss its advantages in terms of information retrieval.

Open SQL Environment: Structured organizational databases, typified by SQL databases at present, are rarely available on the public networks. Instead, their networks are closed, either limited to an organizational environment or proprietary products. They network in two ways, client-server and multidatabase, via middleware. In both cases, knowledge of system catalogues and query language dialects must be hardwired into the application or the middleware platform.

The technical problems of making an SQL database available on an open network environment are analogous to those of making a text database available on that environment. Firstly, the server must have a standard way of making its system catalogues available to the client, of agreeing on the query language dialects and types of data available, and a standard means to verify version compatibility between client and server. Secondly, the query language must support the ability of the server to not only process a query but to retain it as the basis for further queries, in the same sort of way as managing saved result sets in text databases. Finally, the server must support a standard means to manage and account for software actions (such as triggers) originating at a client but residing at a server site. All of these facilities are available in the Z39.50 protocol; the Z39.50/SQL+ project presents an adaptation of that protocol to an open SQL environment.

Z39.50/SQL+: Z39.50/SQL+ can be seen as an extension of the existing Z39.50-1995 (Version3) protocol, uniting the advantages of the SQL RDBMS's with those of Z39.50. This SQL extension is most beneficial (but not restricted to) when a client wishes to retrieve information from an SQL database.

The existing ANSI/NISO Z39.50-1995 communications protocol is an open standard designed to facilitate the search and retrieval of information in databases. The standard specifies formats and procedures governing the exchange of messages between a client and server, enabling a client to request that the server search a database and identify records which meet specified criteria, and to retrieve some or all of the identified records. It includes features which allow the server to advise the client as to the names and characteristics of the elements of the server database and to establish agreement as to what query language is in common between the client and the server. It provides features to manage state during a session, and to manage and account for state across sessions. Resource and access control facilities are also available.

Z39.50/SQL+ introduces a new query type, a type-SQL2 query - a query conforming to the SQL-92 standard which is highly structured allowing search terms and attributes to be specified within the query. Pre-defined attribute sets, which provide a virtual global data schema, are not necessarily required as the SQL server database already stores its metadata within its system catalogues. Like Z39.50, Z39.50/SQL+ still distinguishes two types of response records that may occur from the server: database and diagnostic records. It introduces a new record syntax, SQL2-RS, by which database records may be returned and similarly an additional error format, SQL2-ERR. No changes to the resource report and access control formats are envisaged at this stage. Minor extensions to the Explain record syntax include version and catalogue table name parameters.

Z39.50/SQL+ provides to the SQL user a stateful communication environment with the full flexibility and query power of SQL when connecting their working environment to a remote database. Z39.50/SQL+ clients will be able to formulate complex queries, either by using SQL or one of its derivatives, such as Query-by-Example (QBE). Queries may be formulated on multiple tables supporting cartesian products, unions, intersections, joins on matching columns, and projections on given columns, as well as being able to use powerful constructs for expressing conditions, performing aggregate and comparison operations, partitioning tables into groups and much more. In addition, SQL RDBMS's provide structured, organizational databases complete with data management facilities including system catalogues, flexible indexing and query optimization - providing efficient access and retrieval of both data and metadata.

Project Status: A full technical report, *`Z39.50/SQL+ Project'*, has been written and is available on the web. The building of the prototype, using the YAZ Z39.50 toolkit, began in March. The project is scheduled to have a base-line implementation ready for demonstration in early October, with Explain and Extended Service facilities available in early `97. http://www.dstc.edu.au/DBU/research_news/z3950.html

In Summary: Z39.50/SQL+ presents an open SQL standard to facilitate stateful internet access for controlled information retrieval from remote relational databases. It is the facility to manage and account for state within and across communication sessions that distinguishes Z39.50/SQL+ from both existing SQL communication standards and proprietary middleware.

Univ. Tennessee/Knoxville Position Paper #1

Efficient and Authenticated Sharing and Indexing of Internet Resources

Shirley Browne and Keith Moore, University of Tennessee

Jack Dongarra, University of Tennessee and Oak Ridge National Laboratory

Position paper for WWW Consortium Distributed Indexing/Searching Workshop

Internet communities need to be able to share indexing information between domains and between organizations in order to facilitate interdisciplinary and inter-organizational resource sharing. Organizations and communities need a way to share resources without each organization running a Web crawler that accesses each of the other organizations' Web sites, because such an *n^2* solution does not scale. Resources providers need an easy-to-use mechanism for publishing metadata in a place where users and indexing services can access it easily and efficiently.

Scalable, efficient access to popular resources requires widespread replication, or mirroring, of these resources. With current mirroring schemes, a different name (i.e., URL) is given to each copy of a replicated file. Web crawlers must access all the mirrored copies and deduce which ones are duplicates. A user who accesses a mirrored copy, perhaps after being given a list of alternative mirror sites by an overloaded server, has no way of verifying that the retrieved mirror copy is identical to the original. Thus, there is a need for a single location-independent name for all copies of a file, so that metadata can be attached to this name rather than to the individual copies. This metadata should include a digitally signed file fingerprint so that a user can verify the integrity of a retrieved file copy. There is also a need for users to be able to verify the authenticity and integrity of metadata that comes from different sources.

The Resource Cataloging and Distribution System (RCDS) under development at the University of Tennessee is addressing the above needs. The system components include catalog servers, location servers, and file servers. Resource providers assign location-independent names to resources and submit metadata to an RCDS catalog server. An authorized file server that mirrors a copy of a file registers its name-to-location binding with an RCDS location server. An RCDS catalog server provides a centralized location from which Web crawlers can gather metadata. For clients such as Web browsers, an RCDS catalog server resolves a name to associated metadata, which may includes names for individual files. An RCDS location server resolves a name to a list of locations. The RCDS catalog server design provides for attaching a digitally signature to an assertion or to a set of assertions, where an *assertion* consists of an attribute-value pair.

Ideally, RCDS should use a standard format for assertion metadata, so that it presents a standard interface to clients such as Web browsers and Web crawlers, but no suitable standard currently exists. Text representations of metadata are problematic because of changes introduced by editing and other processing that invalidate a digital signature over the byte contents. The Harvest SOIF format is in practice a text-based format, although it allows arbitrary content for the value of an attribute. A digital signature could conceivably be attached to an entire SOIF record, if the record could be guaranteed not to change during transfer and processing, although this would not allow for selective signing of subsets of assertions. To be suitable for use with RCDS, SOIF would also need to allow a URN for the identifer of an object, in addition to a URL.

Univ. Tennessee/Knoxville Position Paper #2

Towards High-Quality Searching on the Web

Mike Berry and Shirley Browne

Computer Science Department, University of Tennessee

Murray Browne, School of Information Science, University of Tennessee

Position paper for WWW Consortium Distributed Indexing/Searching Workshop

In order to achieve higher quality searching on the Web, there needs to be a shift from the operational goal of "get every file that contains one or more of the keywords I entered, ranked by where and how often they occur" to "retrieve the resources that best satisfy my information need, with the most relevant and highest quality ranked the highest". From the theoretical information science point of view, one would ideally like to have comprehensive high-quality topical indexes and be able to route an information need to the most appropriate of these to search. The problem is that such indexes would be prohibitively
expensive or impossible to construct by entirely manual methods, given the size and diversity of the Web.

Thus there is a need for semi-automated methods that build on already developed Web technologies and assist domain experts in constructing and maintaining high quality topical indexes. The overall Web search engines would then become interfaces to the distributed collection of these specialized topical indexes.

To enable interoperation between search services, it will be necessary to standardize descriptions of query types and search capabilities, and to standardize the syntax for standard query types, so that:

- Search services can advertise what types and capabilities they support
- Clients (be they browsers, applets, agents, or other search engines) can formulate queries in a standard format, and
- Search services can translate from the standard to an internal format

It will also be necessary to characterize both the content and the quality of different search engines and their underlying databases. We believe that clustering techniques based on semantic analysis will provide the most effective characterization of content.

Quality of a search service should be determined by evaluation based on standard performance measures and criteria. Currently used measures consist mainly of the number of items in the database and the speed with which search results are returned, with no evaluation of the relevance of the results to the expressed information need. Measures that approximate recall and precision and that evaluate the accuracy of the ranking of search results need to be developed. These measures could perhaps be based on relevance judgments solicited from users, and on comparison of query results across multiple search services. Comparative ratings of database quality will provide a way to combine ranked results from different search services. Standard performance measure will also help in evaluating new indexing and retrieval methods. We believe that a new generation of statistically based semantic retrieval methods, such as Latent Semantic Indexing (LSI), will provide better performance than the current general of lexical matching methods.

Univ. Washington/Metacrawler Position Paper

Representations of URLs by Web Search Services

Erik Selberg & Oren Etzioni

Current global Web search services, such as Lycos and Alta Vista, are unable to provide comprehensive coverage of the Web. One solution to this has been the use of meta-search services which query each base service, such as MetaCrawler and SavvySearch. While the economic issues of a meta-search site can be resolved amicably and profitably between the meta-service and the base service, there are still some detailed technical issues which should be addressed.

Multi-service search services are able to collate results from many different search services, such as Lycos or Alta Vista. One of the many challenges faced by such meta-services is that each base service represents the contents of its database in a different manner. In order to compensate for this, meta-services must employ a variety of heuristics and custom code in order to collate information in a manner appropriate for users. This is a problematic approach, as it is not robust to changes in the base servers' representations, as well as being a wasteful approach, as often the meta-engine must compute information about the data, possibly by downloading it from a congested network, which the base service could have provided.

Most global Web search services use a confidence score as their only indication of relevance to the user's query. This score is just a number --- most services use natural numbers from [0 .. 1000] with 1000 being a "perfect match." Meta-search engines will typically normalize the score, and rank based upon a summation of that score. This method has problems, in that one service's notion of a "high score" is dramatically different than anothers. For example, given the query "Used Car," one service may give a high score based upon the word "Car" appearing in the title, whereas another will give an equally high score because "Used Car" appears somewhere in the body text.

What is needed is a richer formulation of the results returned by search services. This representation should include things such as:

- Number of terms matched;
- Number of terms matched on complete word;
- Semantic location(s) of matched terms (e.g. in <h2> block);
- Physical location(s) of matched terms (e.g. char 123 out of 1024);
- Unique identifier for URL contents (e.g. MD5 checksum);
- Date added to database;

Further, the ability to obtain information on the metric used for calculating ranks should available, as well as the ability to obtain information as to why particular URLs were excluded. For example, it should be possible to query a search engine with a query text and URL, and ask why the URL wasn't returned with the results of the query text.

These features, and undoubtably others, are needed in order to enable meta-search services to perform as well as they are able. Without this information, meta-search engines either need to infer the data which wastes computation time, download the page and extract the information which wastes network bandwidth, or do without, which produces less than optimal results. The obvious solution is create a standard representation which allows search services to convey the most information about their results to their users, be they human or artificial.

US Geological Survey Paper #1

Proposal for an Information Locator Service

by Eliot Christian, United States Geological Survey

Prepared for the Distributed Indexing/Searching Workshop at MIT, May 28-29, 1996

Information needed to locate other information takes many forms, and no single access mechanism can be optimal for all applications. This proposal is to define a simple and generalized Information Locator Service to be supported in addition to other protocols such as HTTP, Whois++, gopher, and LDAP. Clients could search across compliant servers to obtain all manner of locator information, including the characteristics of other Internet information resources. Even servers that support high-performance applications such as name resolution could separately provide search access to the metadata maintained.

The Information Locator Service would adopt existing international standards such as a minimal subset of those adopted by the Government Information Locator Service (GILS) Application Profile. (GILS itself is in U.S. law and policy at Federal, state, and regional levels; internationally in countries such as Canada, Japan, Australia, and the United Kingdom; and in intergovernmental initiatives such as the G7 Global Information Society.)

The GILS Application Profile, approved internationally in May 1994, adopts some Internet RFC's and a subset of ANSI Z39.50-1995. Z39.50 does begin to address the handling of multi-lingual information, supports various security and other arrangements for fee-based and free dissemination of information, and has been implemented in either a stateless or stateful mode of operation. The GILS Profile defines only the behaviors of compliant servers--clients are unconstrained and can range from simple user interfaces to sophisticated software agents.

There are commercial GILS-compliant servers as well as freeware implementations for all popular server platforms. These GILS-compliant servers are serving many kinds of information resources with wide variation of structure, from HTML to USMARC files, as well as relational and Postgres databases. Gateways exist for Web browser access in addition to standalone or browser add-on clients that use the search protocol directly.

Because the GILS Profile adopts a subset of the ANSI Z39.50 standard, GILS-aware clients can already freely search hundreds of professionally maintained resources such as library and spatial data catalogs collectively valued in the tens of billions of dollars, with much more available on a fee basis. There are also hundreds of WAIS databases freely accessible, and thousands more WAIS databases maintained behind HTTP servers.

A compliant server appears to a client as though holding a searchable set of information locator records. Each locator record can characterize other information of any kind, at any level of aggregation, and includes URI's and MIME types for Internet resources. For example, a locator record that describes another server might include a listing of the words most characteristic of that server's contents and so act as an intermediary resource for information discovery.
Searches can be content-based using full-text searching or other manner of feature extraction. Or, the search may take advantage of structured attributes such as well-known elements and relations, though it is not necessary to have a canonical format for structured metadata. Natively or through gateways, the service can support search of many different metadata structures--HTML, SGML, X.500, SQL databases, PURL's, Handles, Dublin Core, SOIF's, IAFA, Internet mail, DIF's, Whois++ templates, spatial metadata, etc. Whenever appropriate, servers simply map local semantics to registered attributes, and the attribute registry itself is extensible through an established process.

Acknowledgments, **References**, **Topics in Software Implementations**

US Geological Survey Position Paper #2

Advanced Search Facility for Federal Documents

Creation of an Advanced Search Facility for Federal Documents on the Internet requires improvement of existing search engines. This requirement stems from the different user communities such a system must service. The first user community is composed of individuals who may have only a vague idea of what they are looking for. The second community is composed of individuals that are looking for a specific document or piece of information. Both groups require better ranking of result sets to focus the selections presented by these systems.

Most current search facilities do not present the results of a query in a form that is useful to either groups of users. The problem is in the order that documents are presented. Often commentary or discussion about an information resource rather than a pointer to the resource itself is returned. An Advanced Search Facility for Federal Documents must allow the user to specify the ranking criteria to be used to determine the order of presentation. Original documents must have a higher rank that commentary or even "pointer pages."

Metadata such as described in the the Government Information Locator Service (GILS) profile can be used to provide the additional guidance needed to properly rank results returned from a query. The GILS metadata take the form of locator records which can be automatically created by the indexer software then exported for editing by humans. Additional metadata could be added and the locator records served for use by serarch engines in locating and ranking documents. These locator records would add sufficient information to allow the search engine to identify original documents. Existing GILS servers could provide locator records to the system via Z39.50 protocol.

The use of locator records served via Z39.50 protocol will allow other systems to provide information for the ranking of query results. The Federal Geographic Data Committee (FGDC) clearinghouse, for example, could provide information about spatial data and documents because it's application profile has GILS as a subset. Those GILS servers operated by all Federal Agencies would provide an additional source of information. Federal Agencies are required to create and maintain GILS records about their programs. Documents that contain Dublin Core Metadata would assist the automatic creation of the locator records.

An Advanced Search Facility must also understand place and time as searchable attributes. These are necessary to answer a question of the form: "What small business opportunities are available now near Atlanta Georgia." The GILS profile and Dublin Core both support a bounding rectangle that can assist in answering this question. With the minimum bounding rectangle, greater use can be made of existing locator systems like the FGDC Clearinghouse.

An Advanced Search Facility, when built, will gather and index all Federal Government information on the internet in a distributed fashion. Those servers capable of running the indexer will make their indexes available to other systems on the net. The index information will be made available using Z3950 v3 (1995) protocol and perhaps LDAP as well as protocols internal to the system. Recent tests have demonstrated that GILS compliant metadata can be searched and served using LDAP protocol as well as Z39.50.

Verity, Inc. Position Paper

Distributed Indexing and Searching

Nick Arnett

Internet Evangelist

Verity Inc.

(narnett@verity.com)

Prepared for the Distributed Indexing/Searching Workshop

This paper is intended to describe current research and product development at Verity Inc. and to support development of open industry standards for distributed indexing and search on the Internet. Verity developed the first commercially available indexing spider, which the company continues to sell and develop in conjunction with its line of indexing, search and retrieval products. The viewpoints in this paper are subject to change.

The primary goal of open standards for indexing is to acquire data objects across the Internet for efficient indexing and incremental updates of existing indexes. Secondary goals include the desire to reduce server and network loads. Verity's research and product development efforts are focused on merging existing and proposed standard protocols with new, open information gathering protocols. Although the company is not committed to a particular technical direction, it views certain technologies as important antecedents of the information gathering protocol that is to be developed. These include the "robots.txt" standard for robot exclusion (presently supported by Verity products) and the Harvest system developed at the University of Colorado (supported by third-party Verity developers, including the University). However, these antecedents do not address issues that the company believes are critical to today's Internet environment. Furthermore, the Web's primary transfer protocol, HTTP, as well as its antecedents, FTP and Gopher, are inefficient for Internet index maintenance operations, except as carrier protocols.

For example, the "robots.txt" exclusion could be enhanced to include greater information about documents and collections of documents stored on a server. There is no standard means of storing and obtaining meta-information such as titles and owner/maintainer identification for document groups. The "robots.txt" file or similar resource descriptions could accomplish this with an extensible set of well-known descriptions of such data.

The Harvest Summary Object Interchange Format (SOIF) was an important step forward in the effort to transfer large amounts of new and changed information with "push" and "pull" mechanisms, which are critical to efficiency. Verity believes that SOIF can be a building block in an open information gathering protocol that would be stronger than SOIF in terms of incremental updates and generation of data objects specific to the requirements of a robot.

Finally, Verity favors negotiation-based protocols, conceptually similar to those used in recent modem communications, in which the pair of communicators falls back to the most efficient commonly supported protocol. The least common denominator would be today's typical robot operation -- a series of GET or HEAD requests. The most sophisticated protocols would include "push" and "pull" requests, compression negotiation and search query-based index updates (which would take advantage of a search engine's ability to return results based on field data such as last-modified date). The protocols would allow the commuicators to exchange data objects and deletion lists for index maintenance.

Vivid Studios Position Paper

Index and Search Position Paper

```
Title:     Index and Search Position Paper
Company:   vivid studios
Version:   1.0  May 6 1996
Author:    Christian Mogensen
```

Vivid's concerns fall into three areas: interoperability, communication, and internationalization.

Interoperability

The lack of a standard conveying the context of a web document has led to the use of keywords embedded in comments or META tags and similar workarounds. Unfortunately, meta-data for non-HTML data is not stored and the early work on ALIWEB and Harvest self-indexing has not caught on as much as hoped. This has led to the evolution of mega-indexers like Inktomi, Altavista, and Infoseek. These indexers share no data nor do they cooperate with sites when it comes to generating index data.

A standard summary or index format and collection point would make it easier for indexers to download an entire website's document collection. As a result, bandwidth and compute resources would be used more efficiently since indexes would hit websites only once.

The solution then is to devise a more generic meta-data format that will let both HTML and non-HTML files be indexed and catalogued. The PICS format is one possibility. The important thing is to agree on either one standard that encapsulates the set of annotations or on a meta-standard that would allow gateways between various formats and/or annotation types.

Communication

A web server is optimized towards serving a single document at a time but an indexer wants collections of documents to work with. Either we need to come up with a special URL to allow indexers easy access to collections or we need to introduce a new service geared to the needs of indexers. Harvest's broker network is a step in the right direction but the Summary Object Interchange Format (SOIF) needs the flexibility of PICS. The current crop of NSF-funded digital library projects can have a large bearing on this discussion.

Internationalization

Indexing engines are only now becoming sufficiently HTML/SGML aware such that they can resolve entity references before storing documents in their repository. More work is needed to deal with simple things such as accented letters and varying character sets. A document may exist in multiple language versions, all of which may exist under the same URL depending on the `Accept-Language` headers that are sent. Meta-data is required to describe the dimensions on which a document may vary. (This meta-data could be sent as HTTP headers, for example.)

In summary, there needs to be more cooperation between indexers and document servers in order to make better use of scarce resources. Content providers have to provide more meta-data as servers and document publishing systems become more complex. Exposing this data to the world in a standard way will add tremendous value to the document collection and ultimately make information more accessible and useful.

Xerox Corp. Position Paper

Result Merging in Distributed Indexing

Jan Pedersen and Hinrich Schuetze, Palo Alto Research Center, Xerox Corporation

Distributed search systems should be able to return ranked result lists as well as the unranked result sets that are common in Boolean systems and Harvest/Glimpse. Ranking is difficult in a distributed environment because simply combining results across servers is known to be inferior to the ranking that would be achieved if the documents where centralized. This is because ranking strategies make use of collection statistics which, in the case of a meta-collection composed of a number of distributed collections, are not available to each individual server. Some systems try to circumvent the need for statistical weighting by ranking according to the number of term matches (e.g. WAIS). However, it is well-known in the information retrieval literature that unweighted searches perform worse than weighted searches [1]. To address this problem, we believe that two-way communication of collection statistics must be included in standards for effective distributed searching.

Most term weighting methods in information retrieval rely on the collection document frequency of a term, i.e. the number of documents in the full collection in which that term occurs. When a user creates a meta-collection on the fly by selecting a number of collection servers to search over, a term's document frequency for the meta-collection can be computed by summing the document frequencies from the individual collections. The protocol should therefore enable the client to obtain the document frequencies of the current query terms from the individual collections. However, unless that information is communicated back to each individual server, the individual servers will not have the information necessary to form optimal weights. Instead they must rely on local statistics, which produce inferior results to global statistics [2]. The protocol should therefore allow for the communication back to each server of global collection statistics to be used in weighting. Finally, in order for the scores from different servers to be comparable, the same scoring function has to be used by each server. We propose that the most common weighting functions be specified in the standard and that the protocol allow the client to select which weighting function to use on a particular search. The following tf.idf formula is an example of one of the common weighting function in information retrieval:

sum over all terms i shared by query and document:
log(N / N(i)) * square-root(tf(i,document)) * square-root(tf(i,query))

where N is the total number of documents in the (meta-)collection, N(i) is the document frequency of term i, tf(i,document) is the term frequency of term i in the document, tf(i,query) is the term frequency of term i in the query and both document vector and query vector are assumed to be cosine-normalized.

In summary, we argue that results from information retrieval research motivate an extension of the protocols between clients and servers to support effective multi-site ranked searches. The standard protocol should allow clients

- to obtain characteristic statistics about each server's collection
- to select from a standard set of scoring functions and
- to send term statistics on the current meta-collection to a server (along with a particular query)

[1] Salton and Buckley. Term-weighting Approaches in Automatic Text Retrieval. IP&M Vol. 24, No. 5, pp. 513-523, 1988.
[2] J.P. Callan, Zhihong Lu and W. Bruce Croft. Searching Distributed Collections with Inference Networks. SIGIR 95, pp. 21-28, 1995.