Workshop A: Web−wide Indexing/ Semantic Header or Cover Page

Co−Chairs: Bipin C. Desai, Brian Pinkerton

Dr. Bipin C. Desai
Concordia University
Montreal, Canada
Email: bcdesai@cs.concordia.ca
Messages: (514)−848−3040
Fax: (514)−848−8652

Brian Pinkerton
Department of Computer Science
University of Washington
FR−35
Seattle, WA 98195
Email: bp@haole.cs.washington.edu
Fax: (206) 522−2768

Workshop A: Web−wide Indexing/ Semantic Header or Cover Page

Co−Chairs: Bipin C. Desai, Brian Pinkerton

Half−day workshop on Monday, April 10, 1995
9.30 – 13.00

The amount of publicly available information resources on the Web is increasing rapidly. As this trend continues, finding these resources becomes more difficult. Several systems, notably Archie, Jumpstation, Lycos, WebCrawler, RBSE Index, and Harvest have attempted to solve this problem by building indices and allowing users to search them.

These Web−wide indexing solutions do not, in practice, share information with each other. In addition, the databases often fail to provide users with specific or complete answers to their queries.

This workshop has two goals: to find ways that index providers can share indexing and Web−structure information, and to explore ways to improve the query experience for users. Both goals stem from the need to address the increasing scale of the Internet: as the size of the problem increases, we need to be more efficient at building indices, and better at focusing on what the users are looking for. As we find solutions to these problems, it will enable us to build more efficient, consistent, and powerful indices of the World−Wide Web.

In addition to a general discussion of Web−wide indexing, the workshop will have two specific tasks:

1. to examine some of the existing protocols for sharing information to see what we can use and what we need to build, and
2. to envision an operational plan for putting these tools to use on an experimental basis.

Allowing indices to cooperate and exchange information can help solve several problems. First, it would allow the index builders to do a more efficient job of indexing. Indexing information for Europe might be collected in Europe, then transmitted in bulk to the United States. Or, we may find ways to build a distributed index, and avoid even the bulk transmission. Second, it would enable different retrieval engines to run against the same set of indexing information, providing better service for users and opportunities for research on different kinds of retrieval. Finally, it would server as an experimental tool for learning about decentralized indexing.

This workshop is not meant to set any standards for indexing or exchange of indexing information. However, if it serves as a starting point for the experimentation that will give us the experience necessary to propose standards in the future, should that be desirable.

The workshop environment is the ideal place to do this work; we will bring together people with experience building and running Web−wide indices. The ideal participant would be involved in building or operating an Internet information discovery system, or an expert in the field of database systems, distributed computing, expert systems, information retrieval, or library studies.

References

De Bra, P., Houben, G–J., & Kornatzky, Y.
Navigational Search in the World–Wide Web
Desai, Bipin C.
Semantic Header aka Cover Page
Fletcher,J.
Jumpstation
Koster, M.
ALIWEB (Archie Like Indexing the WEB)
McBryan, Oliver A.
World Wide Web Worm
Pinkerton, Brian
Finding What People Want: Experiences with the WebCrawler

Workshop A: Web−wide Indexing/ Semantic Header or Cover Page

Co−Chairs: Bipin C. Desai, Brian Pinkerton

Workshop A:

Web−wide Indexing
Semantic Header or Cover Page

Summary

What to index?

- Use the anchor term used for the HTML links,
- Use the title and headings of the HTML page,
- Use the full text to create an index,
- Use the filename of the HTML resource,
- Word occurrence − URL pairs.
- Inverted indices of keywords
- Indexes of interesting keywords

How are index created?

- Use a robot to scan the Web for new and changed HTML resource
- Server side support based systems,
- Different frequency of updates

What is indexed?

- Approximately twenty search engines with accompanying services for parts of the WWW,
- Each covers a part of the Web,
- Gateways to other indexing services such as WAIS

What is needed?

- Establish a Web Indexers' Working Group,
- Hierarchical searching,
- Share information and avoid replication,
- Parallel, fault−tolerant, and scalable index server,
- Language(Natural) independent
- Find all the _____ having concept/property −−−−−−,
- Index other form of resources: images, graphics or sound,

- Capture structure of the Web, hyper−media,
- Common interface,
- Index generation with revision control

Dr. Bipin C. Desai *Concordia University*
Montreal, Canada
Email: *bcdesai@cs.concordia.ca*
Messages: *(514)−848−3040*
Fax: *(514)−848−8652*

The Semantic Header and Indexing and Searching on the Internet [1]

(c) Bipin C. **Desai**
Department of Computer Science
Concordia University
7141 Sherbrooke St. W
Montreal, H4B 1R6
bcdesai@cs.concordia.ca
http://www.cs.concordia.ca/~faculty/bcdesai/

Keywords:Bibliographic record, Content description, Database Systems, Expert System, Indexing, Searching, URC

Abstract

This paper describes an indexing system called semantic header for Internet resources. The semantic header contains the meta−information for each "publicly" accessible resource on the Internet. It also describes the registering system and the distributed database representing the union catalog of resources on the Internet. This database would be used in a search system to facilitate search.

Introduction

The trend in most research institutes, universities and business organization to interconnect their computing facilities using a digital network has become the accepted method of sharing resources. Such networks, in turn, are interconnected allowing information to be exchanged across networks using a common interchange protocol(viz TCP/IP). The number of such interconnected networks (Internet) continues to grow and with the emergence of powerful workstation−based servers connected to these networks, it is possible to support local as well as the remote search and retrieval of information stored on any component of the interconnection. At this time a number of information sources, both public(free) and private(available for a fee), are available on the Internet. They include text, computer programs, books, electronic journals, newspapers, organizational, local and national directories of various types, sound and voice recordings, images, video clips, scientific data, and private information services such as price lists and quotations, databases of products and services, and speciality newsletters.

There is a need for the development of a system which allows easy 'search for and access to' resources available on the Internet. It has been observed that distributed information systems, even though under control of a single administrative unit, create multiple problems typically caused by differences in semantics and representation, incomplete and incorrect data dictionaries (cataloging) [DESA4]. These problems would be magnified manyfold in any distributed information system which tries to integrate the resources offered by information systems over the Internet. It is important, also, to avoid problems encountered in a library system where, in spite of the fact that while the same cataloging system[2] is used, the same item may be differently catalogued/classified in two different libraries.

Such problems could be avoided by starting with a standard index structure and building a bibliographic system using standardized control definitions. Such definitions could be built into the knowledgebase of an expert system based index entry and search interfaces. Furthermore, there must be a mechanism to revise index information as the resource changes over time. Finally, annotation of a resource by independent users should be allowed.

The bibliographic entry system should be distributed and accessible to providers as well as users of the Internet. In a distributed system such as the Internet, it is natural to have the providers of resources, prepare and enter the bibliographic information about each resource using the standardized index scheme. The entry system should be a distributed system and the index should be recorded in a distributed database. Finally, a search system to help in locating and retrieving appropriate information with ease from this database is required.

Whereas the bibliographic entry and search systems (clients) could be located locally at the providers and users of information resources respectively, the bibliographic database system(server) should be distributed and replicated at a number of regional nodes for enhanced availability and response. The entry and search systems have to be supported by an easy−to−use graphical interface for entering the index information and access to it. These systems should incorporate the expertise and knowledge of expert cataloguers and reference librarians with help system to guide the user at all steps. The search system, should in addition provide appropriate feedback indicating the number of hits for each search, and help in providing access to the relevant resources. The navigation of database and resource nodes and the protocols and filters used would be selected by the system, thus facilitating the task of the user. The purpose is to provide uniform access to all resources, as is done in the centralized information system through the intermediary of an expert system analyst. The overall structure of such a system is given in Figure 1

Source of Information and Meta−Information

Information sources can be classified into three categories[KATZ]: primary, secondary and tertiary. Primary information is the original material in the form of published or posted articles, monographs, reports, dissertations, programs, images, movies, etc. Other primary sources such as personal communications are not usually available. Secondary sources, sometimes called meta−information, are used as indices to these primary sources of information and are created after a delay which may be a few months to a few years. The meta−information is data about the primary source. A tertiary source of information is a combination of selected and distilled information

from primary and secondary sources.

The purpose of indices and bibliographies (secondary information) is to inventory the primary information and allow easy access to it. Preparing a bibliography requires finding the primary source, identifying it as to its subject, etc., describing it for later matching for unknown future users and classifying it according to accepted norms.

Since an index is to be used by many users, it has to be accurate, easy to use (usage via author, title, subject, etc.) properly classified, up−to date and complete for its area of coverage. In order for a bibliography to be useful, it must fill a real need. The success of Archie as a bibliography system (for files available on the Internet via FTP) is that it provides a simple interface to users who are aware of the name of a program, file or the general nature of the file likely distributed from one or more anonymous FTP sites. In the case of the on−line bibliography to the Internet resources such as the Web, the need is for the system to be current within a short period (minutes or at most hours) of the posting of a new resource. Compare this with the bibliography system for printed publication which requires weeks or months in the case of the on−line databases, longer for the CD version and upto years for the printed version. Even the on−line database needs a considerable amount of time before documents are indexed in bibliography.

The method of compiling a traditional bibliography varies. At one extreme, we have scholars spending years of their lives evaluating sources and compiling annotated and descriptive entries for each item. The accuracy of this bibliography is high but the coverage tends to be limited. At the other extreme, we have the semi−automatic mechanism which scans the published works from limited sources (by domain, language, or geographic regions) and assigns each work to appropriate sub−subject(s). Access from multiple headings may be provided. This is desirable because an item may deal with more than one topic. Whereas the bibliography prepared in the former method could be more accurate it tends, however, to be retrospective rather than current.

The dependence on titles as a search criterion dictates that they must be indicative of the contents of the document. This is not always the case hence someone (the author or the cataloger) has to add annotation, keywords or key phrases to indicate the actual content. Accuracy or quality of a document can be indicated by including reviewers' opinions. However, such opinions are rarely accessible to the cataloger. Another feature of importance to the user of an index, is the presence of an accurate abstract. An abstract provides a summary of the material and thus is more indicative of the contents than the title or keywords supplied by the author, bibliographer or selected from scanning the text. Reference librarians and library users tend to use such annotated bibliographies to help choose among competing sources.

Features such as division of the bibliography by subject and sub−subjects, though of concern in the manual systems, should not be apparent in the electronic form. However, access through these criteria must be supported. Weeding of bibliography entries, which are for Internet resources no longer accessible, though attractive may require careful thought from the point of completeness. The archiving of resources in central libraries could mean that such weeding of the bibliography would not be necessary.

A Cataloging and Searching System

Library catalogs are prepared by a specialist and for each entry, it records the author, title, publisher, place of publication, date of publications and other details. The term union list, in library lexicons, is used to refer to the catalog which is the union of the catalogs of a number of participating libraries. It indicates which item is located where. In this sense, the bibliography, forms a union list of all sources of documents. Since the item in question is not in electronic form, it requires the intermediary of the inter−library loan mechanism to borrow it (usually from the nearest location which permits the title to be borrowed or if possible to photocopy sections of it.)

Currently a large number of documents exist in addition to the files whose names could be searched via systems such as Archie or Xarchie. The popularity of the World Wide Web[BERN, BERN3] and browsers such as Mosaic[MOSA] has prompted many researchers to start publishing on−line. Attempts to provide easy searching of relevant documents has lead to a number of systems including WAIS, and more recently a number of Spiders, Worms and other creepy crawlers.[DEBR, FLET, KOST, MCBR, META, THAU, SEAR, WEBC, WWWW]

However, the problem with many of these indices is that their selectivity of documents is often poor. The chances of getting correct documents and missing relevant information because of poor choice of search terms is large. In addition, the user is required to access the actual resource, based on just the title and author information, as is provided through a library catalog, and decide whether the resource meets the needs.

These problems are addressed in our proposed system by using an appropriate index entry called Semantic Header[BCD2] and providing a mechanism to register, manage and search the bibliography. The system is an *active* system requiring the provider of information to register the resource by entering an index entry for the resource. Since the provider is responsible for preparing the index entry, there is the potential for its accuracy to be high.

The overall system uses knowledge bases and expert sub−systems to help the user in the register and search process. One such need for an expert system is in avoiding chaos introduced by differences in perception of different indexer. Hence, some form of standardization of terms used has to be enforced. We envisage this through the intermediary of an expert system based engine. The index generation and maintenance sub−system uses the knowledge and expertise of the expert cataloguer to help the provider of the resource select correct terms for items such as subject, sub−subject and keywords. Similarly, another expert system is used in the search sub−system to help

the user in the search for appropriate information resources. The third component of the system is a distributed and replicated database of the bibliography to resources available on−line. The database is in the background and the users are not aware of its presence much less of its distributed and replicated nature. These components are described below.

Semantic Header

The heart of any bibliography or indexing system is the record that is kept for each item that is being indexed. Standardization of a bibliographic entry allows libraries to exchange information about their collections. A number of projects in the Library domain have addressed the problem of cataloging and in particular cataloging of information in electronic and multi−media format. CORE[CROM], MARC system[BRYN, CRAW, MARC, PETE], MLC[HORN, ROSS, RHEE] and TEI[GAYN, GIOR] are examples of some of these initiatives. These existing and proposed indexing systems range from a minimum to full level of bibliographic information. However, such systems are designed for professional catalogers and many of the items included in them, though useful, are beyond the comprehension of most providers or users of information.

We have proposed a simple index structure called Semantic Header [DESA2] for resources accessible directly on the Internet. The structure of the index is similar to the ones used for most libraries indices and include other information deemed useful for on−line systems. The syntax of the semantic header is the HTML markup language[BERN2] which is based on the SGML markup language. However, the user working with the index entry system is guided through the process by an expert system. This system guides the user in the choice of standardized terms through an easy to use graphical interface.

We give, in Figure 2 below, the structure of the Semantic Header. An example of use of the semantic header is given in Figure 3. The intent of the semantic header is to include those items that are most often used in the search of an information resource. Since the majority of search begins with a title, name of one of the authors (70%), subject and sub−subject (50%)[Katz], we have made the entry of these items to be mandatory in the semantic header. The abstract and annotations are useful in deciding whether the resource would be useful; these items are also included. Logically the entries in the semantic header are not positionally sensitive. However, for ease of use, we have arranged the fields in Figure 2 using the traditional library catalog layout.

The first field of the semantic header is the title of the resource. It is a required field and is given within the tags beginning with <title> and terminated by </title>. The next field is a alt−title and is used to indicate a secondary title or an alternate title of the resource. This field is optional. The subject and the sub−subject of the resource is indicated in the next field which is a repeating group (a multi−part field with one or more occurrences of items in the group). All resources must have at least one occurrence for this field.

The character set used and the language of the resource is given in the next two optional fields. The author of the resource is given in the next repeating group. The sub−fields are for name, organization, address, phone and fax numbers and e−mail address. All sub−fields except the name are optional except where the author is an organization in which case the organization must be given. The term author is used to include the role of programmer, creator, artist, etc.

The list of keywords is included by a field marked by the tags <Keyword> ... </Keyword>. Each resource must have at least one keyword. If a published version of the resource is available, this is indicated by the next field which is followed by a place of publication and appropriate publication code (code name and number) such as ISBN followed by a number.

The dates of creation(required), expiry and update, if any, are given next. The version number, if any, the intended coverage and the security or distribution classification is indicated in the next three fields.

The location (URL[BERN1]) of the item is indicated by the next field indicated by the tags <URL> ... </URL>. It could include a list of one or more locations where the item may be available. The URN[RFC1737] field gives the unique name of the item, if any. This name may be used instead of a location (URL) if the item is likely to move or may be accessible from multiple locations[3].

The semantic header contains an entry for an archive site. The field UAS (Universal archive site) is used to indicate the archive site for the resource. It is expected that the resource will exist at this site beyond the expiry date of the resource, if any. Of course, the site itself is guaranteed to exist beyond the life of any resource. It is envisaged that the archive site could be an independent resource provider. One example of such a traditional resource provider is the national library in most countries. One possibility is for the national libraries such as the Library of Congress in U.S., British Library, National Library and CISTI in Canada, to archive Internet resources. However, private, for profit, corporations could be alternate sites for archiving resources. Archiving would provide an anchor for the otherwise ephemeral nature of some resources on the network.

The abstract and annotations are given in the next fields. The abstract is provided by the author of the resource; the annotations are made by independent users of the resource. The annotation cannot be modified and includes the identity of the user along with a digital signature.

List of hardware and software required is included in the semantic header as a repeating group. This is followed by the size of the resource and the cost of accessing it[4].

The last set of items in the semantic header is the control items such as the account to which credits are to be made for charges for accessing the resource, encoded

passwords or the digital signature of the provider of the resource. Any change to the updatable part of the semantic header requires the password or digital signature. Another control piece of information is the digital signature of the resource itself. This may be used to authenticate the resource when it is retrieved through a semantic header. It is assumed that there is a mechanism to access the resource's digital signature.

<semhdr>
<title> required </title>
<alt−title> OPTIONAL </alt−title>
<Subject> a list each of which includes fields for subject and up to two levels of sub−subject: at least one entry is required </Subject>
<char−set> character set used: OPTIONAL </char−set>
<language> of the information resource: OPTIONAL </language>
<author> required
a list each of which includes name, organization, address, etc. of each person/institute responsible for the information resource: at least the name or the organization and address is required </author>
<Keyword> required: a list of keywords </Keyword>
<Publisher> OPTIONAL in case of a published version </Publisher>
<PublPlace> OPTIONAL in case of a published version </PublPlace>
<Code>OPTIONAL in case of a published version </Code>
<Dates>
<Created> required: </Created>
<Expiry> OPTIONAL </Expiry>
<Updated> system generated </Updated>
</Dates>
<Version> OPTIONAL: version of the resource </Version>
<Coverage> OPTIONAL: nature of the resource </Coverage>
<Classification> OPTIONAL: security level of the resource </Classification>
<URL> A list of locations (URL) Unique Universal Resource Locator/Call No for this resource: at least one required </URL>
<URN> unique name of the resource (URN) </URN>
<UAS> site where the item is to be archived </UAS>
<Abstract> OPTIONAL but recommended </Abstract>
<Annotation> OPTIONAL </Annotation>
<SysReq> OPTIONAL: list of requirements in hardware and software
<Hardware> OPTIONAL: list of hardware required </Hardware>
<Software> OPTIONAL: list of software required </Software>
</SysReq>
<size> size of the resource in bytes </size>
<Cost> OPTIONAL: cost of accessing the resource </Cost>
<control>
<Ac> account number </Ac>
<password> required: encoded password or digital signature of provider of resource for initial entry and subsequent update </password>
<signature> digital signature of the resource for authentication </signature>
</control>
</semhdr>

Figure 2. Structure of the Semantic Header

<semhdr>
<title>Semantic Header and Indexing and Searching on the Internet</title>
<alt−title>Sailing the Internet with a navigational System</alt−title>
<Subject>

<General>Computer Science </General>
<Sublevel1>Information Storage and Retrieval</Sublevel1>
<Sublevel2>indexing</Sublevel2>

<General>Library Studies</General>
<Sublevel1>cataloging</Sublevel1>
<Sublevel2>semantic header</Sublevel2>

<General>Computer Science </General>
<Sublevel1>Artificial Intelligence</Sublevel1>
<Sublevel2>expert systems</Sublevel2>

<General>Computer Science </General>
<Sublevel1>Database Management</Sublevel1>
<Sublevel2>distributed databases</Sublevel2>

</Subject>
<Language> English </Language>
<Character> ISO−8879 </Character>
<author>

<aname>DESAI, Bipin C.</aname>
<aorg>Concordia University, Department of Computer Science</aorg>
<aAddress>7141 Sherbrooke Street West, Montreal, QC, CANADA, H4B 126
</aAddress>
<aphone>(514) 848 3025</aphone>
<aFax>(514) 848 8652</aFax>
<aemail>bcdesai@cs.concordia.ca</aemail>

</author>
<Keyword>

Bibliographic record
Content description
Database Systems
Expert Systems
Indexing
Searching
URC

</Keyword>
<Dates>
<Created> 1994–07–11</Created>
<Expiry>1995–08–07</Expiry>
<Updated>1995–02–07</Updated>
</Dates>
<Version> 1.0 </Version>
<Coverage> Universal </Coverage>
<Classification>Public </Classification>
<URL> http://www.cs.concordia.ca/~faculty/bcdesai/cindi–system–1.0.html</URL>
<URN><comment>Unique Universal Resource Name for this resource. No such service exists to date. In the absence of one, we use the concatenation of Title, first author, first subject creation date and version number. Do we really need another level of complexity especially if we have a good index and catalogue system? Is the current system of using domain name followed by other names not good enough? It is the most distributed version possible. Here domain names not only signify Internet domain but other domains such as ISBN, UPC, etc. </comment>
Semantic Header and Indexing and Searching on the Internet|Computer Science|Information Storage and Retrieval|indexing|DESAI, Bipin C.|1994–07–11|1.0
</URN>
<UAS><comment>Universal Archive Site where this document is archived</comment> ftp://ftp.cs.concordia.ca/bcd/cindi–system–1.0.html</UAS>
<abstract>This paper describes an indexing system called semantic header for Internet resources. The semantic header contains the meta–information for each "publicly" accessible resource on the Internet. It also describes the registering system and the distributed database representing the union catalog of resources on the Internet. This database would be used in a search system to facilitate search.</abstract>
<Annotation></Annotation>
<size> 44000 </size>
<Cost><comment>Cost, Currency<comment> 0.27, Can$</Cost>
<control>
<Ac> BCD's Swiss number a/c </Ac>
<password> thequickbrownfoxjumpsoverthelazydog </password>
<signature> 01001010101110101101010110011101 </signature>
</control>
</semhdr>

Figure 3 An example of a Semantic Header Entry

Index Registering Sub–system

The index entry and registering sub–system provides a graphical interface (Figure 4) to facilitate the provider (author/creator) of a resource to register the bibliographic information about the resource. The interface allows the provider to enter the information and it provides help by means of pop–up selection windows and an expert engine (not shown in Figure 4) to suggest controlled terms. Once the information is correctly entered the author can decide to register the Semantic Headed entry in the Semantic Header database. When the header information is accepted by the database, the author/creator is notified. A password or a digital signature is to be provided when the semantic header is first registered and for all changes made to it. Since the encoded password or digital signature is not accessible by anyone other than the original registrar of the index entry, the entry can only be updated by person(s) who are cognizant of it. Changes that may be made could be due to changes made in the resource or its migration from one system to another. A copy of the semantic header is stored at the site of the resource. It is desirable that the semantic header be attached to the actual resource. However, this can not be done until all hardware and/or software systems can handle such a header (viz. ignore it).

The system verifies the accessibility of the resource being added. Also the digital signature of the resource is retrieved and added to the semantic header. The purpose of this last piece of information is to establish the veracity of the resource when it is retrieved through a semantic header. If the resource is corrupted, this veracity validation would fail and the user would be notified; no charges, if there are any, would be made.

If each resource is given a unique name (URN), the semantic header database can be used for mapping from URN to URL. Since only one semantic header could be associated with a given URN, a search with a given URN will retrieve at most one semantic header. One of the URLs in it can be used to access the resource in question. This form of search can be implemented at a low level without the need for a graphical interface.

The index entry that is registered is communicated to a database described below.

The Semantic Header Distributed Database System

The index entries registered by a provider of a resource is stored in a distributed database system (SHDDB). From the point of view of the users of the system, the underlying Semantic Header database may be considered to be a monolithic system. In reality, it would be distributed and replicated allowing for reliable and failure–tolerant operations. The interface hides the distributed and replicated nature

of the database. The distribution is based on subject areas and as such the database is considered to be horizontally partitioned [DESA5].

It is envisaged that the database on different subjects will be maintained at different nodes of the Internet. The locations of such nodes need only be known by the intrinsic interface. A database catalog would be used to distribute this information. However, this catalog itself could be distributed and replicated as is done for distributed database systems.

The Semantic Header information entered by the provider of the resource using a graphical interface is relayed from the user's workstation by a client process to the database server process at one of the nodes of the SHDDB. The node is chosen based on its proximity to the workstation or on the subject of the index record. On receipt of the information, the server verifies the correctness and authenticity of the information and on finding everything in order, sends an acknowledgment to the client.

The server node is responsible for locating the partitions of the SHDDB where the entry should be stored and forwards the replicated information to appropriate nodes. For example, the semantic header entry of Figure 3 would be part of the SHDDB for subjects Computer Science and Library Studies.

Similarly the database server process is responsible for providing the catalogue information for the search system. In this way the various sites of the database work in a cooperating mode to maintain consistency of the replicated portion. The replicated nature of the database also ensures distribution of load and ensures continued access to the bibliography when one or more sites are temporarily nonfunctional. The performance of search with the growing size of the SHDDB database could be improved by using techniques used in databases[DESA6].

The Search System

The guiding principle of the design of the search system uses the model of a human reference librarian. S/he is called on to help in identifying the best sources of information for a given purpose and to aid in the selection of materials to meet a particular interest or need. The reference librarian seeks the responses to these queries by using information derived from bibliographic search processed through the librarians own expertise and knowledge of the relevant subject. In addition, users of a library have access to the same bibliographic indices and many of the information databases from which they are called on to select relevant titles or weed out irrelevant ones.

A typical query to a reference librarian can be divided into two categories: known and unknown[KATZ]. In the former, a user asks for an item identified by author, title, or publication source. In the latter, the need of the user is fuzzy; s/he has no idea of any of the identifiers of the needed item. Even in the case of the known queries, there is the possibility that the user may have the wrong author, right author but the wrong title, wrong dates or incorrect volume number or issue number for a serial. It may also happen that even when these are correct, the item is not the one that meets the need of the user.

A specific search and research type query may require the user to peruse a number of titles and select from among them. This type of query involves users who have fuzzy notions of their needs and their questions are vague. They involve a certain amount of trial and error retrieval of documents and their browsing.

One problem that human librarians deal with is that of the inability of the users to ask the relevant questions. The reference librarian, through a dialog with the user tries to narrow down the user's needs in terms of what and how much information is required. In many cases the librarian is called upon to match the user needs with the sources of information. For example, an article from the popular press may be appropriate for a lay person as opposed to one appearing in a prestigious journal dedicated to the subject.

In the search component of the proposed system we plan to incorporate the expertise used by a reference librarian. This expertise will guide the user in entering the various search items in a graphical interface similar to the one used by the index entry system (Figure 5). The expert search sub−system requires the expertise of a reference librarian to be built into it to help users formulate queries and launch these queries. As in the case of the index generation sub−system, the expert system provides help in choosing appropriate search terms for index entries such as subject, sub−subject, keywords etc. The expert system, for optimization of search, uses the following type of statistics for a typical use of a bibliography[Katz]

−70% of the queries is by use of titles, or by author's name

−50% of queries start with subject and it tends to be complex requiring subdivision and refinement.

The search system also uses a graphical interface and a client process. Once the user has entered a search request, the client process communicates with the nearest SHDDB catalogue to determine the appropriate site of the SHDDB database. Subsequently, the client process communicates with this database and retrieves one or more semantic headers. The result of the query could than be collected and sent to the user's workstation. The contents of these headers are displayed, on demand, to the user who may decide to access one or more of the actual resources using a graphical window as in Figure 6. It may happen that the item in question may be available from a number of sources. In such a case the best source is chosen based on optimum costs. The client process would attempt to use appropriate hardware/software to retrieve the selected resources.

Annotations and Reviewing

The scientific world depends on peer review of documents submitted for publication. Such annotation used for reviews tend not to be published. However, comments to the editor made by readers of the serials are usually published and are accessible to the community. Since many of the resources on the Internet tend to be non−reviewed, it would be useful for a user to have access to annotations made by other users for a given resource. The proposed system allows users to add annotations to an existing resource. These annotations are stored along with the index in the SHDDB.

The annotation sub−system is similar to the indexing subsystem. However, only a few of the indexing entries, to uniquely identify the resource in question, are required (Figure 7). An annotation made by any user can be entered and would be registered with the identity and digital signature of the user. Each annotation could than be incorporated in the index entry (at least logically) and could be retrieved with the index. Such annotations, by recognized persons would be a valuable guide for future users.

The peer reviews of electronically submitted papers could be implemented using such annotations. Authentication of reviews has to be done by an appropriate editorial board.

Conclusions: Advantages of the approach

Current index systems are based on harvesting the network for new documents and such documents are retrieved and their contents used to provide terms for the index. The big disadvantage with his scheme is the unreliability of the index entries produced and the lack of an authentic abstract for the item. Currently, such schemes are relevant for Web text documents and are not applicable to other resources. Another problem with this approach is the unnecessary traffic on the network and lack of cooperation and sharing among different systems. Finally, the unfeasibility of this approach as more and more providers of information would require payments. Creating an index would require payment. Furthermore, users, without having a better idea of their contents, would not be inclined to retrieve resources which, from their titles, seem irrelevant.

In the proposed system, the provider of the resource is the one who prepares the index information. Consequently, such index entry would be more reliable than the one derived by a third party or by simply scanning a document. The presence of an abstract affords the provider of the resource to give a pertinent abstract or summary. Such a summary in the index allows users to make better informed decisions regarding the relevance of the source resource.

The system provides an expert system−driven graphical interface for the provider of the resource to produce an index entry, and have this entry entered in the index database. The expert system provides help in choosing appropriate terms for index entries such as subject, sub−subject, keywords etc. It also is responsible for verifying the consistency of the index entry and accessibility of the resource and then posting the index entry to the index database.

In addition, the index database contains a number of control entries for the resource. Control entries are items such as size of the resource, the password for authenticating subsequent updates of the index entry, and a list of annotations made about the resource by independent users

Acknowledgment

The author wishes to gratefully acknowledge the many thought provoking comments by colleagues Carol Caughlin, Lee Harris and Rajjan Shinghal. This work was supported in part by a grant from the Seagram Funds for Academic Innovation.

References

[BERN] Berners−Lee, T., & Cailliau, R., "WorldWideWeb: Proposal for a HyperText Project" http://info.cern.ch/hypertext/WWW/Proposal.html

[BERN1] Berners−Lee, T. "UR* and The Names and Addresses of WWW objects", http://info.cern.ch/hypertext/WWW/Addressing/Addressing.html see also RFC 1738,

[BERN2] Berners−Lee, Tim, Connolly, "Hypertext Markup Language, Internet working draft", http://info.cern.ch/hypertext/WWW/MarkUp/HTML.html

[BERN3] Berners−Lee, T. "Wide Web Initiative: The Project",

[BYRN] Byrne, Deborah J., "MARC manual: understanding and using MARC record", Libraries Unlimited, Englewood, Colo. 1991.

[CRAW] Crawford, Walt, "MARC for Library Use: Understanding USMARC", G. K. Hall, Boston, MA, 1989.

[CROM] Cromwell, Willy, "The Core Record: A New Bibliographic Standard", Library Resources and Technical Services, Vol. 38−4, pp. 415−424, 1994.

[DEBR] De Bra, P., Houben, G−J., & Kornatzky, Y., "Search in the World−Wide Web", http://www.win.tue.nl/help/doc/demo.ps

[DESA1] Desai, Bipin C., "WebJournal: Visualization of Web Journey", August 1994, http://www.cs.concordia.ca/WebJournal.html

[DESA2] Desai, Bipin C., "Cover page aka Semantic Header", July 1994, http://www.cs.concordia.ca/semantic−header.html, revised version, August 1994, http://www.cs.concordia.ca/~faculty/bcdesai/semantic−header.html

[DESA3] Desai, Bipin C., Shinghal, Rajjan, "A System for Seamless Search of Distributed Information Sources", May 1994,

http://www.cs.concordia.ca/w3−paper.html

[DESA4] Desai, Bipin C., Pollock, Richard, "MDAS: A Heterogeneous Distributed Database Management System", Information and Software Technology, January 1992, Vol. 34−1, pp. 28−41.

[DESA5] Desai, Bipin C., "An Introduction to Database Systems", West, St. Paul, MN 1990.

[DESA6] Desai, Bipin C., "Performance of a Composite Attribute and Join Index", IEEE Trans. On Software Engineering, Vol. 15−2, pp. 142−152, 1989.

[FLET] Fletcher, J. 1993., Jumpstation, http://www.stir.ac.uk/jsbin/js

[GAYN] Gaynor, Edward, "Cataloging Electronic Texts: The University of Virginia Library, Experience", Library Resources and Technical Services, Vol. 38−4, pp. 403−413, 1994.

[GIOR] Giordano, Richard, "The Documentation of Electronic Texts Using Text Encoding Initiative Headers: An Introduction", Library Resources and Technical Services, Vol. 38−4, pp. 389−401, 1994.

[GNAM] Global Network Academy Meta−Library, http://uu−gna.mit.edu:8001/cgi−bin/meta

[HORN] Horny, Karen L., "Minimal−level cataloging: A look at the issues− A symposium", Journal of Academic librarianship, Vol. 11, pp. 332−334.

[KATZ] William A. Katz, "Introduction to Reference Work", Vol. 1−2 McGraw−Hill, New York, 1987

[KOST] Koster, M., "ALIWEB(Archie Like Indexing the WEB)", http://web.nexor.co.uk/aliweb/doc/aliweb.html

[KOST1] Koster, M., "Simple Unified Search Interface (SUSI)", http://web.nexor.co.uk/susi/susi.html

[KOST2] Koster, M., "Configurable Unified Search Interface", http://web.nexor.co.uk/public/cusi/cusi.html

[MARC] Library of Congress, "MARC manuals used by the Library of Congress", American Library Association, Chicago, 1969.

[MCBR] McBryan, Oliver A., "World Wide Web Worm", http://www.cs.colorado.edu/home/mcbryan/WWWW.html

[MCBR1] McBryan, Oliver A., "GENVL", http://www.cs.colorado.edu/home/mcbryan/public_html/bb/summary.html

[META] Experimental Search Engine Meta−Index, http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Demo/metaindex.html

[MOSA] NCSA Mosaic, http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html

[PETE] Petersen, Toni, Molholt, Pat (ed), "Beyond the book: extending MARC for subject access", G.K. Hall, Boston, MA, 1990.

[POST] Post, R., "Lagoon: a WWW cache", http://www.win.tue.nl/lagoon

[RFC1357] "A Format for E−mailing Bibliographic Records", D. Cohen.: can be obtained via anonymous FTP from anyone of: ds.internic.net, nis.nsf.net, src.doc.ic.ac.uk, munnari.oz.au and a number of other site.

[RFC1737] "Functional Requirements for Uniform Resource Name", K. Sollins, L. Masinter: pl. see RFC1357 above.

[RFC1738] "Uniform Resource Locators(URL)", T. Berners−Lee, L. Masinter, M. McCahill: pl. see RFC1357 above.

[ROSS] Ross, Rayburn M., West, Linda, "MLC: A contrary viewpoint", Journal of Academic librarianship, Vol. 11, pp.334−336

[RHEE] Rhee, Sue, "Minimal−level cataloging: Is it the best local solution to a national problem?", Journal of Academic librarianship, Vol. 11, pp.336−337, 1986.

[SEAR] Search WWW document full text, http://rbse.jsc.nasa.gov/eichmann/urlsearch.html

[TAYL] Taylor, Arlene G., " The information universe: Will we have chaos or control?", American Libraries, Vol.25−7, pp. 629−632, 1994.

[THAU] Thau, R., "SiteIndex Transducer", http://www.ai.mit.edu/tools/site−index.html

[WEBC] WebCrawler, http://www.biotech.washington.edu/WebCrawler/WebQuery.html

[WWWC] World Wide Web Catalog, href=http://cuiwww.unige.ch/cgi−bin/w3catalog

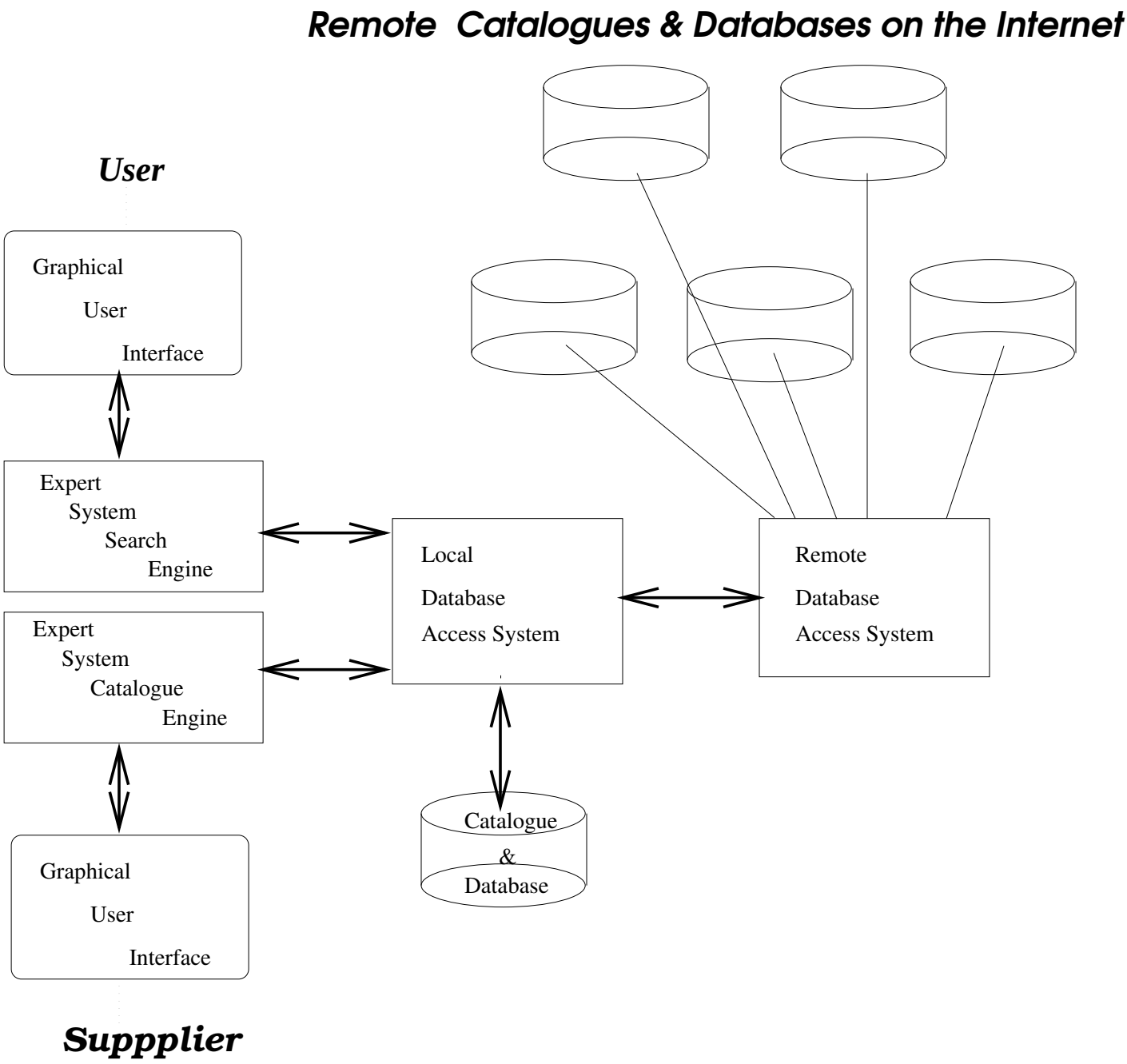

Figure 1 Overall Structure

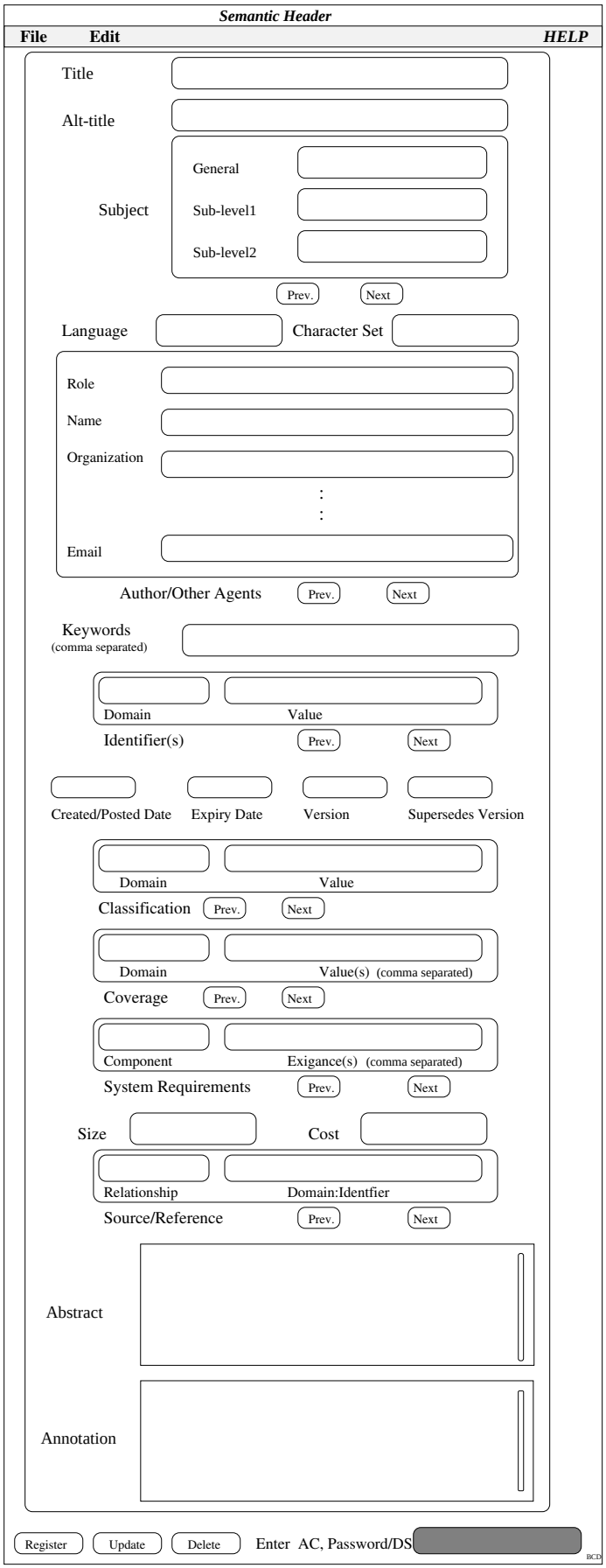

Figure 4 Graphical interface for entering/updating Semantic Header **BCD**

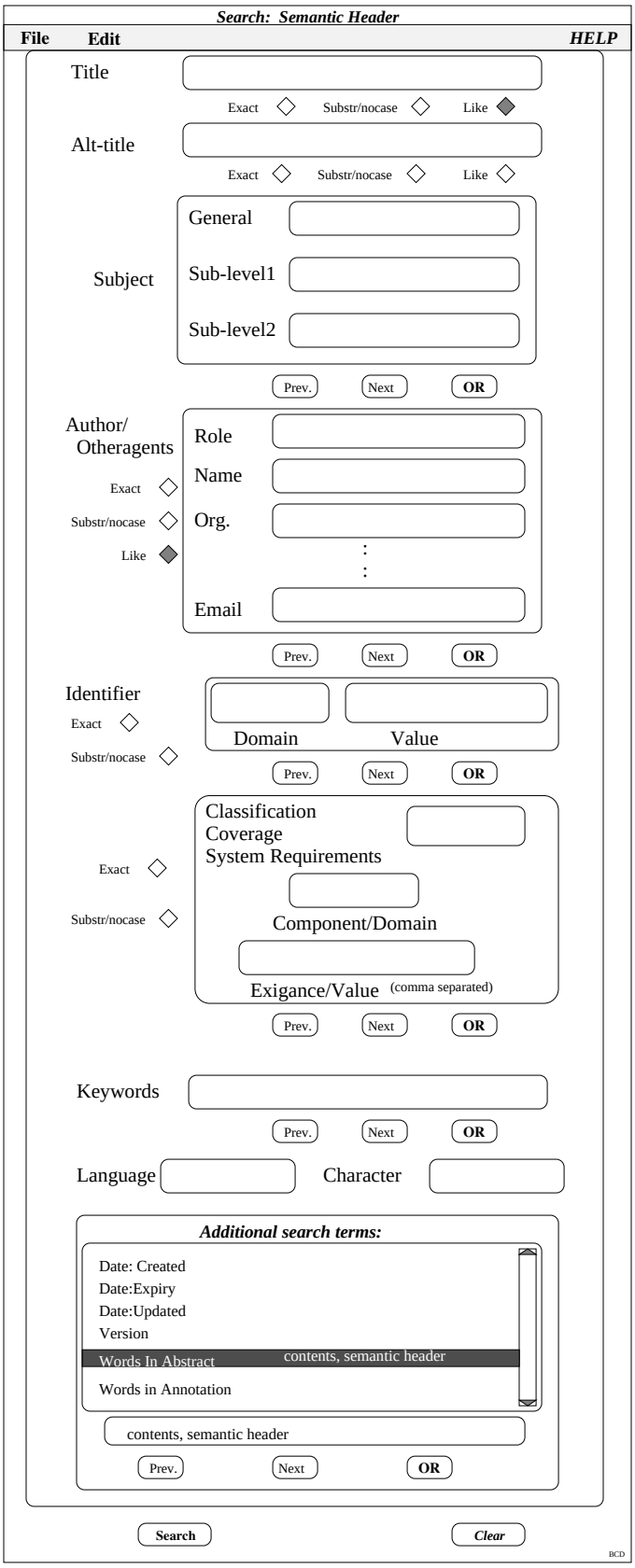

Figure 5 Data entry for searching Semantic Header(s)

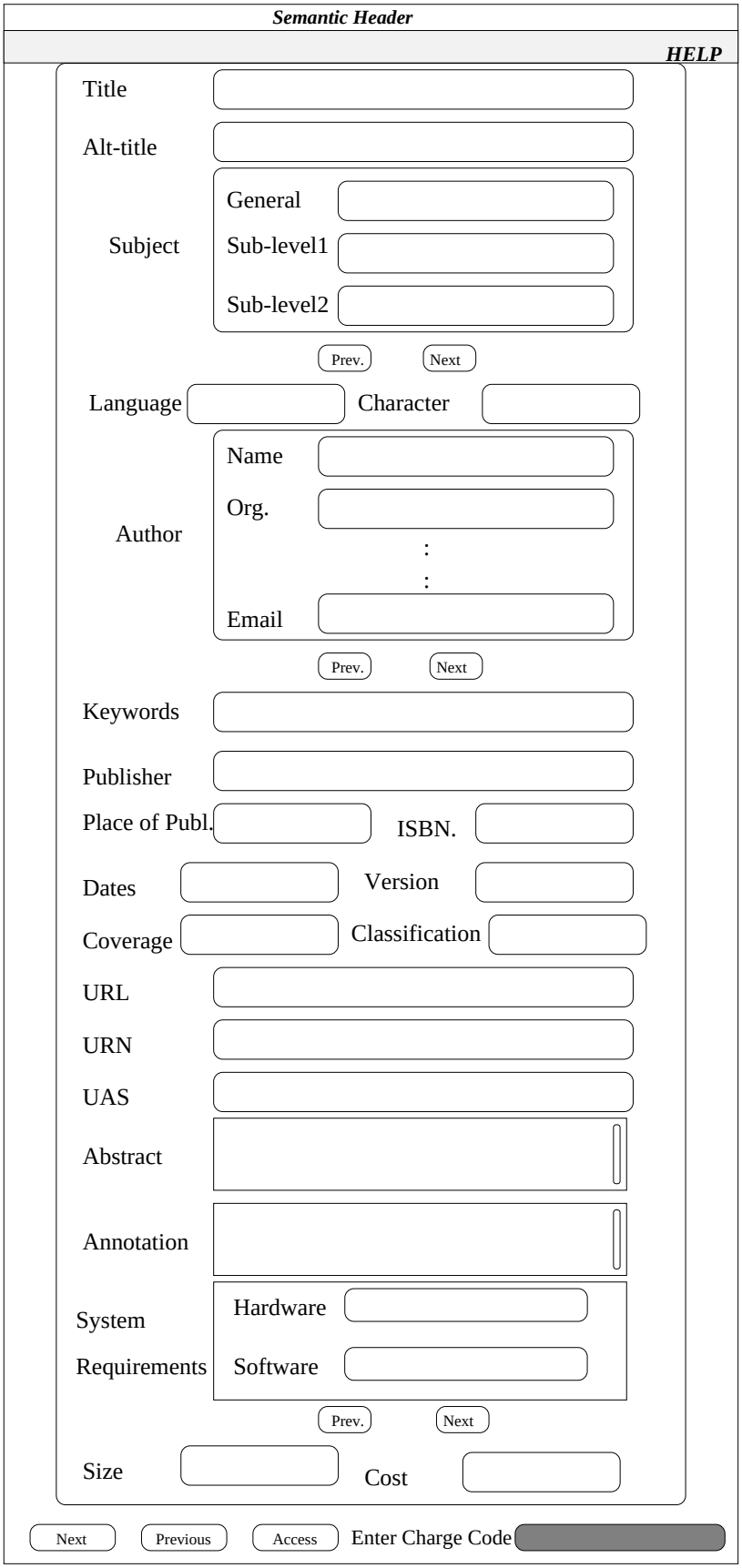

Figure 6 Display format of Semantic Header

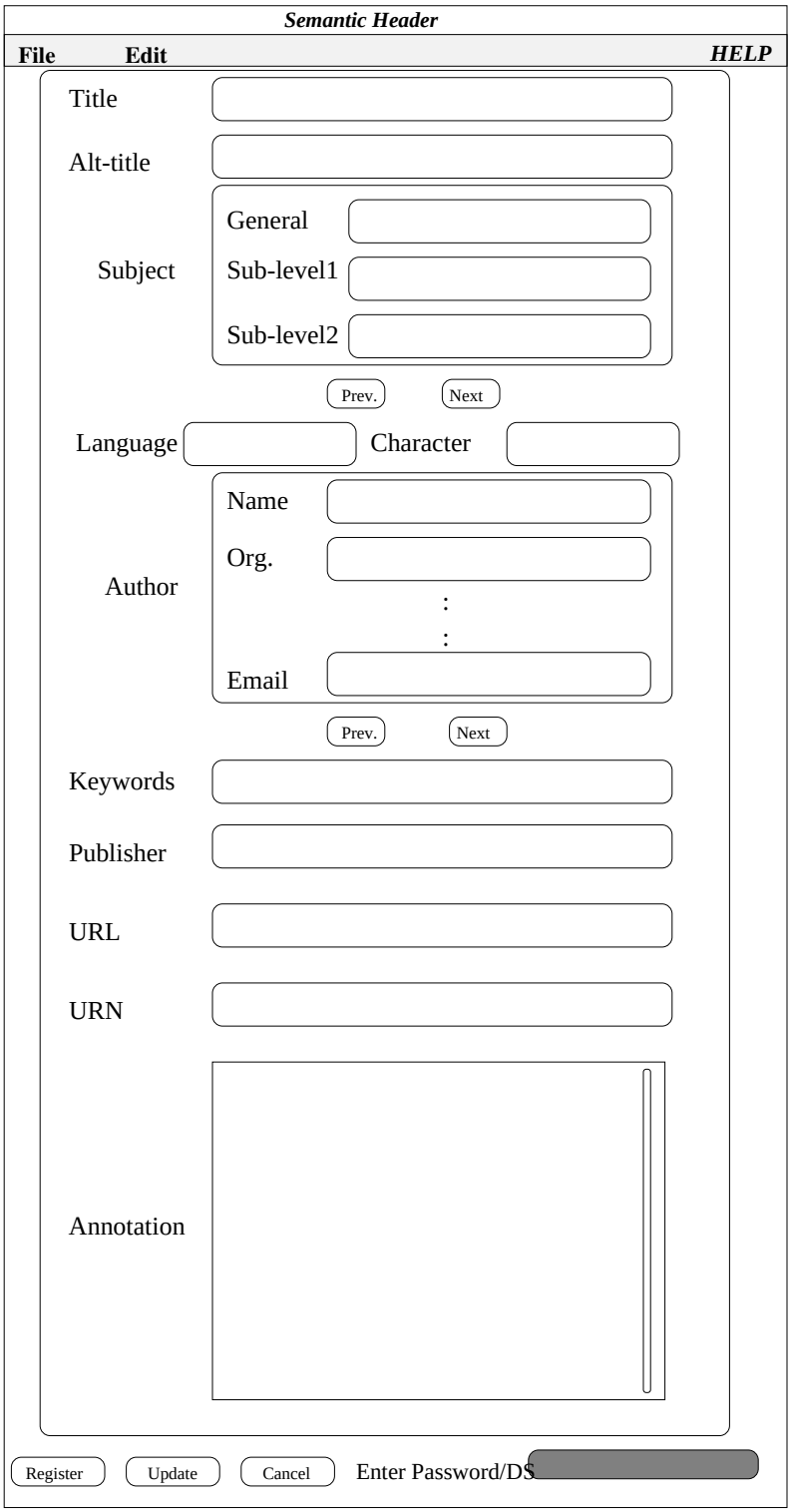

Figure 7 Annotation of a Semantic Header

[1] This paper describes the CINDI subsystem – a part of the CUILT Project for Developing a Virtual Library Prototype.

[2] Libraries use a number of basic catalogue systems such as s Library of Congress, Dewey Decimal and MARC. Even among MARC there are slight differences as in LCMARC and CANMARC.

[3] The idea of the semantic header is to provide bibliographic information about resources and by including both the URN and a list of URLs it also provides a mapping from URN to URL.

[4] Such costs could change over time and require updating.

Workshop A: Web–wide Indexing/Semantic Header or Cover Page

Chair: Bipin C. Desai, Brian Pinkerton

The Open Text Web Index, and General Issues
in Web Indexing

Tim Bray

Introduction to the Open Text Web Index

Open Text will be offering a free general-purpose WWW search facility, whose provisional name is "The Open Text Web Index." The Web Index is being built now, and
will go live and public in the immediate future. It will be usable, if perhaps not generally
announced, at the time of the workshop.
The functions provided by the Web Index will not be surprising to anyone familiar with a
service such as Lycos or the Webcrawler. As do they, it uses a robot to scan the Web for
new and changed pages, builds an index to the text of those pages, and uses the index to
provide search functions in a standard HTML/HTTP way to standard Web clients.
Perhaps the most important distinguishing feature of the Web Index is its motivation. It is
a private sector marketing project, not a research exercise. It will be based on Open
Text's commercial search engine, and it will run on servers sited on the Internet backbone,
hosted by UUNET Canada, who will do facilities management. I am a co-founder of
Open Text, and my personal reason for doing this is because it's fun. I sold this to my
management and to UUNET purely as a marketing exercise; the goal is to make
something really useful and thus show our products and UUNET's services off to a few
thousand people a day.
Here is an overview of what the Web Index provides:

Search access to every word of the full text of the indexed WWW pages.
Good bandwidth, based on its adjacency to the Internet backbone.
Search, with no performance loss, by words, partial words, numbers, or phrases.
Support for ISO-Latin encoded texts, with correct handling of accented European
characters.
Complete multi-term Boolean and proximity search.
Ranked search, based on user supplied weights and a choice of weighting algorithms.
Search optionally limited, per search term, to Title, Address (URL), H1, Anchor; and
eventually more HTML structures if there is a demand for it; the cost of doing so is
basically zero.
High currency; the Index will be updated nightly without ever going off-li
Smooth growth; because of the way the Open Text engine works, as the size of the
Index and the number of its users grows, it will expand to run in parallel on multiple
servers, transparently to the user.
SGML capability; the Open Text search engine provides full support for ISO-standard
SGML. Currently, there is little "real" SGML available on the Web, but if m
arrives, the Web Index will be able to take advantage of it to provide increased power
and flexibility in search and display operations.
Note that the Web Index, unlike, for example, Lycos, makes no effort to ap
semantic/linguistic judgment to the relevance of terms. It simply finds matches to
strings, words, and phrases - very efficiently.
There is nothing particularly interesting from a Web-hacker's point of view in all this. I
use an httpd that has been modified to dispatch to the search engine directly; clearly the
cgi-bin mechanism would be a bottleneck at the kind of load levels such a system needs to
support. I use some vile tricks to pseudo-parse the so-called HTML enough to support
some structure in searching. The spider is moderately clever in how it traverses the Web.
Almost everything is in perl. In general, given that I have a budget to work with, I have
been willing to deal with problems, where reasonable, by brute-force application of CPU
power and disk acreage.
The Open Text indexer and searcher are applied pretty well as they come straight out of
the can. Given that we have customers running databases approaching 100 GB, it should
hold up under the strain for a while anyhow.
Futures and Issues for Discussion
In the long term, a monolithic index of the whole Web probably cannot fly, given its
scope, volatility, and growth. Thus, if its users are to escape the current Hansel-and-
Gretel practice of navigation by trailing cookie crumbs, some sort of distributed indexing

architecture is required. There seems to be an overwhelming case that the architecture
should be analogous to that of the Internet and the Web: peer-to-peer, with only interface
protocols specified, and nothing at all specified until it can be shown to have worked.
The work done by the Harvest project seems to point in one valuable direction; also the
Z39.50/WAIS family of retrieval protocols have at least the virtue of being well-
understood and having lots of working software around.
I personally don't think we have the right solution in hand at the moment, though. My
worry is based on our experience, first in the Oxford English Dictionary Project, then at
Open Text, Text with distributed, highly structured document databases,
The problem is that of structure. For now, it's not a big issue because HTML is laughably
primitive in its structural expressiveness, even it were being used properly. However: (a)
HTML is growing and becoming more expressive, (b) there is some possibility of people
putting real SGML out on the Web, and (c) in many cases, the HTML is merely a delivery
format for some better-structured underlying repository. For all these reasons, we'd like
to have a distributed indexing/search protocol that at least has the potential of being
extended to handle complex nested, grammar-defined structured such as those of SGML.
As a first step, it should at least have the power of SQL. A research group at Waterloo
has proposed a very small set of extensions to SQL that equip it for handling
recursive/hierarchical structures like those of SGML; it may well be of interest.
There is also room for discussion of HTTP enhancements (an are-you-there message,
conditional retrieval) which could vastly decrease the load that robots place on the net.
Proposals
I propose that we establish a Web Indexers' Working Group, membership limited to those
who are actually running indexes, and others nominated by them, to start attacking these
problems. I suspect that progress in this area would be of more genuine benefit to the
Web than all the HTML extensions and slick browser features put together.

Workshop A: Web–wide Indexing/Semantic Header or Cover Page

Chair: Bipin C. Desai, Brian Pinkerton

The Norwegian UNINETT Indexing Project

Jeremy Cook

This talk will outline the Norwegian UNINETT Indexing Project. The aim of this project is to provide a national scheme for indexing WWW documents within Norway. The project aims to take existing indexing/searching applications and adapt them to the requirements of the project. Some of these requirements are listed below:

- Hierarchical searching
- Limited (specified) searches, to avoid returning too many references to the same or closely related documents in the same hierarchy
- Integration into existing systems to provide an upgrade path and reduce maintenance requirements.

We are looking for other sites with indexing applicatinos that would be interested in collaborative work in this area.

Workshop A: Web–wide Indexing/Semantic Header or Cover Page

Chair: Bipin C. Desai, Brian Pinkerton

Chris Dodge

My name is Chris Dodge, and I work as a postdoc. research scientist in the computer centre of the Alfred Wegener Institute for Polar and Marine Research (AWI) in Germany. Part of my job here has been to set up our WWW server (at http://www.awi-bremerhaven.de/), and in my role as "Webmaster" here, I've become interested in Web indexing and resource discovery.

Many scientists at the AWI are somewhat sceptical about the benefits that the Web can provide for the academic community, and in some quarters here, the take up of Web usage has been very slow. Part of the problem is that the apparent signal to noise ratio is too low when people try and find information relevant to their field. In an attempt to help our scientists track down interesting information, I have setup a database of documents related to polar, marine and global change research, based on automatic scanning of our proxy server cache. I am presenting a paper on this at the conference, titled:

> "Web Cataloguing Through Cache Exploitation and Steps Towards Consistency Maintenance".

One problem with any list or catalogue of Web resources is that changes on the Web can mean that the lists/catalouges can become out of date. Part of my work includes attempts at the creation of mechanisms to prevent this.

My current attitude towards this work is that while it has been reasonably successful, by itself, it does not appear to provide a very complete list of information on the Web related to polar and marine research. I think that maybe this cache scanning mechanism, in combination with other searching mechanisms, would provide a more complete database.

A further problem is in the identification of appropriate Web documents; in the AWI database it is estimated that at least 27% of the database entries are of no relevance to polar and marine research.

Workshop A: Web−wide Indexing/Semantic Header or Cover Page

Chair: Bipin C. Desai, Brian Pinkerton

Semantic Levels of Web Index Interaction(1)

Position Statement for the Web−wide Indexing Workshop at WWW Spring '95

David Eichmann

Repository Based Software Engineering Program
University of Houston −− Clear Lake
2700 Bay Area Boulevard
Houston, TX 77058
eichmann@rbse.jsc.nasa.gov

Table of Contents

1 −− Introduction

Operators of Web indexing facilities are faced with two dramatic drivers: the explosive growth of the shared artifact that they attempt to index (the Web) and the increasing expectations placed upon them by their clientele. In [3], I posed among other criteria, that service agents should attempt to be authoritative, that is, up−to−date and as complete as reasonable regarding their covered domain. Satisfying the expectations of the user community and the criteria of authoritativeness leads inevitably to the need for indexer/providers of the Web to share information and avoid replication where feasible. Note, however, that I am not advocating the position that research and experimentation in this area is no longer of interest − far from it! I am instead seeking to suggest that the era of "the index with the most URLs cataloged wins" is past and that a new phase of serious research into scalable distributed indexing is needed.

2 −− A Strawman Architecture to Support Scalable Shared Indexing

My thesis is that there must be a hierarchy of semantic structure associated with shared Web indices. Such a hierarchy can allow for easy entry costs and gradual increases in sophistication by an indexer/provider, rather than expecting a substantial development effort in order to 'play.' Any such expectation would result in significant lack of participation, effectively obviating the purpose of the initiative. This section lays out a strawman architecture involving increasing levels of sophistication regarding the information that an indexer/provider could both absorb and serve. Each of them can be configured in (at least) two distinct ways:

- total dump of index entity (e.g., a set of word occurrences) and the URL as pairs
- a single pair of the set of all index entities and the URL

2.1 −− (text, url) pairs

This level entails simple matching of word occurrences to URLs. This is scalable in a number of different ways:

- respond with all occurrences
- respond with X most frequent occurrences
- respond with relevant occurrences
- respond with X most relevant occurrences

The semantic support provided by this level of interaction is minimal there is no

context for the document (i.e., the document's local link neighborhood) or correlation to the inquirer's domain of interest (if any). However, even the simplest of spiders can support this interaction to some degree.

2.2 –– (construct, text, url) triples

This level is a slight variation of that in section 2.1, adding the HTML construct in which the text occurs. This would allow, for instance, WWWW [7] to accept a feed from the RBSE Spider [2] or WebCrawler [9], and only request and/or store title strings, etc.

2.3 –– (header, url) pairs

This level is independent of those in sections 2.1 and 2.2, in that only the HTML header and the URL would be exchanged. This mode would allow exchange of the new proposals for embedded metadata [1] and potentially handle AliWeb–style interchanges as well [6].

2.4 –– Structural Web fragments

The RBSE Spider's current public index is really two distinct databases, one containing word occurrence – URL pairs and one containing a relational representation of the Web as discovered to date. We're currently testing a new implementation that integrates the structure and text into a single data model supporting mix of structural, temporal and textual search criteria. This implies an architecture level based upon retrieval of a local neighborhood of HTML artifacts (e.g., multi–file documents, technical report series, etc.) without the need to interrogate the provider site.

2.5 –– Semantic Web fragments

Enhancing the structural model of section 2.4 through the attribution of nodes with information from sections 2.1, 2.2, or 2.3 yields a enriched layer in the architecture capable of supporting substantial search algorithms and intelligent agents [3, 4].

2.6 –– An ontology–based meta–web

This layer involves the construction of a knowledge representation based, distributed conceptual model of artifacts accessible through the Web, effectively forming a meta–Web comprised of formal characterizations of the Web itself using a shared ontology. The Knowledge Interchange Format (KIF) under development by the ARPA–sponsored Knowledge Sharing Effort [8] is an example of the type of notation that might be used between indexer/providers interacting within such a framework.

References

[1] Desai, B. C., Semantic Header aka Cover Page, http://www.cs.concordia.ca/~faculty/bcdesai/semantic–header.html

[2] Eichmann, D., RBSE's URL database, http://rbse.jsc.nasa.gov/eichmann/urlsearch.html

[3] Eichmann, D., "Ethical Web Agents," *Second International World–Wide Web Conference: Mosaic and the Web, Chicago, IL, October 18–20, 1994, pages 3–13.*

[4] Eichmann, D., Sulla A User Agent for the Web, http://ricis.cl.uh.edu/agents/sulla.html

[5] Fletcher, J., Jumpstation, http://www.stir.ac.uk/jsbin/js

[6] Koster, M., ALIWEB (Archie Like Indexing the WEB), http://web.nexor.co.uk/aliweb/doc/aliweb.html

[7] McBryan, O. A., World Wide Web Worm, http://www.cs.colorado.edu/home/mcbryan/WWWW.html

[8] Neches, R., The Knowledge Sharing Effort, http://www–ksl.stanford.edu/knowledge–sharing/papers/kse–overview.html

[9] Pinkerton, B., Finding What People Want: Experiences with the WebCrawler, http://webcrawler.cs.washington.edu/WebCrawler/WWW94.html

Footnotes

(1)This work has been supported by NASA Cooperative Agreement NCC–9–16, RICIS research activity RB02.

Workshop A: Web–wide Indexing/Semantic Header or Cover Page

Chair: Bipin C. Desai, Brian Pinkerton

Luke Emery

First Step Research is a private commercial enterprise doing work
in the area of public access (Moscow/Pullman) and commercial networking
services. More broadly, we are a consulting organization focused on
the application of advanced technologies to the solution to real
world problems....(pretty broad, and more clearly articlulated
within our www sites: http://www.fsr.com and/or http://www.moscow.com).

Among other things, we are providing an experimental web-based
community oriented information repository with interesting
and relevant local and global stores of information. It is
called Palouse Net and includes a community calender of events,
classified ads, a local business listing, and many other things.
This service is being provided for free to the local web community
and everywhere else. Check out http://www.fsr.com/pn.html. There
are currently about 4000 entries in our repository.

In addition, we are actively pursuing other web based technologies for
both our outside consulting work and internal uses. We have
implemented our own http server, browser (still minimal) and a web
robot that may be used for both internal page and link verification and
scanning the web for information. It contains hooks to be used
for context sensitive information scanning and may be constrained
in many different ways. It does not however, follow ALL the reccomended
robot guidelines ;(.

So, in summary, FSR is quite interested in the evolution of the web,
information repositories, and emerging standards for indexing
and distributed information retrieval, and has a technical background
in networking and distributed computing to back it up.

Workshop A: Web–wide Indexing/Semantic Header or Cover Page

Chair: Bipin C. Desai, Brian Pinkerton

Berkeley Search Engine (BSE)

Paul Gauthier
gauthier@cs.berkeley.edu
http://http.cs.berkeley.edu/~gauthier/

Introduction

A number of Internet/WWW search facilities are available at the current time, all suffering under tremendous load. Response times and failure rates are being driven up by the popularity of and demand for such services. The Berkeley Search Engine is an attempt to cope with these problems through implementation of a parallel, fault–tolerant, and scalable server. This new service will run on a network of workstations (NOW) utilizing the CPUs, memory, and disks of a large collection of workstations. The ultimate goal of this research is to identify and build the tools and framework needed to construct generally useful parallel servers. Fault tolerance and incremental scalability are primary goals.

Details

An Internet search engine is a particularly good application to explore needs for parallel server development. Existing searching services offer evidence that there is a very high demand, one that a parallel server could potentially satisfy. It is important that NOW servers be capable of operating with existing protocols and software through standard communication mechanisms. Building a server which conforms to the HTTP protocol will provide a diverse collection of potential clients from many platforms. The HTTP protocol in particular is fairly well suited as an initial test project for NOW server development. The protocol is simple, stateless and allows for some interesting options for dynamic load balancing.

The architecture of the BSE server will be a collection of workstations joined by a high–speed ATM interconnect. The server as a whole must have a single external contact point (so that a URL may be published for a single machine which doesn't change over time). This special machine's purpose will be to redirect incoming query requests amongst the collection of workstations which will actually conduct searches. The front end machine will monitor the status of the collection of workstations and redirect queries to those with the lightest loads. It should have little trouble attaining very high connection throughput due to the simplicity of its task and the very short duration of connection.

The collection of workstations will maintain collective fault–tolerant data structures to aid in query processing, and will stripe the database across their disks. In response to varying traffic load the size of the workstation pool can be varied by releasing workstations or acquiring idle workstations from the NOW. By aggressive cooperative caching techniques and fault–tolerant distributed data structures a highly efficient database query system will be produced.

Implementation Status

At this point the software to build and maintain the database structure is complete, as is a highly optimized query kernel. A parallel implementation on a NOW is currently underway, using the Split–C language and non–fault–tolerant distributed data structures. Implementation of fault–tolerant distributed data structures is also underway, building directly on top of the Global Unix (GLUnix) layer of the NOW project.

Database Content

A successful Internet search service results from two parts: a fast search engine, and a rich database. As well the ability to perform efficient searches, one requires that the database be rich and have wide coverage of the information resources of the Internet. Our research is concerned with the former of these two needs, producing a searching solution scalable to high user loads and large databases.

Population of the database with documents from the WWW, FTP sites and other network sources has not yet begun. It is our hope that projects such as the Harvest system and other established search systems will begin to make their database content available for exchange. The task of crawling the web for content is best done by a small number of parties who can coordinate their activities and reduce impact on HTTP servers and network load.

By acquiring shared data, it would permit our efforts to be more closely focused on the task of building scalable servers. A division of effort between indexing and data collection researh would be beneficial to both groups.

About this document ...

Berkeley Search Engine (BSE)

This document was generated using the **LaTeX**2HTML translator Version 95.1 (Fri Jan 20 1995) Copyright © 1993, 1994, Nikos Drakos, Computer Based Learning Unit, University of Leeds.

The command line arguments were:
latex2html `-dir /home/orodruin/j/grad/gauthier/public_html -split 0 -address gauthier@cs.berkeley.edu BSE.tex`.

The translation was initiated by Paul_A Gauthier on Thu Mar 23 16:07:11 PST 1995

---

gauthier@cs.berkeley.edu

Workshop A: Web–wide Indexing/Semantic Header or Cover Page

Chair: Bipin C. Desai, Brian Pinkerton

Ted Hardie

NASA provides an enormous amount of information via the World Wide Web: astronomical imagery, current program announcements, archival data, technical reports, educational and technical resources, and, yes, even pictures of the space shuttle. This wealth of data means that walking the web at NASA can produce many unexpected epiphanies.

Unfortunately, a user may have to rely on epiphany; finding any specific piece of information can prove to be a daunting task. Each NASA center maintains its own web servers, and many divisions, branches, and projects choose to publish directly to the web. Subdivisions may or may not be linked hierarchically, and those hierarchies are, in any case, none too clear to those not embedded within them. Subject linkages often recursively cross-link, which can mean that a search leads the user endlessly from pages containing primarily pointers to other pages containing primarily pointers.

One of the NAIC's efforts to enhance access to NASA resources has been a project to examine how users traverse the webspace at the NASA Ames Research Center. In order to conduct this research, modifications were made to the NCSA httpd's logging functions to create a more session-oriented view of web accesses; the access patterns shown by the logs were then analyzed1. This (ongoing) research attempts to understand the basic patterns of use by those who walk the NASA or Ames web from a recognized homepage; how arriving in NASA webspace at a point not perceived as an entry point influences usage patterns; the different usage patterns associated with graphical and non-graphical web browsers; and the search strategies employed by users of indexed reference sources. What follows is a collection of observations drawn from our research which may be of interest to this group; the analysis and data collection are still going on, however, so it should be understood that these observations are subject to later revision.

Preliminary data indicate that the Ames web best serves users with graphical browsers who are interested in the work at a center or division level; the graphical links at the Ames homepage and other major entry points gives ready access to these resources. Users of non-graphical browsers seem to find the same links more difficult to follow, apparently because many of the cues for which links are appropriate are contained in in-line graphics or imagemaps2. (The non-graphical alternatives often simply say "image" or give some equally unhelpful phrase, making it difficult for the non-graphical browsers to follow the links.)

Most pages at Ames have appropriate backlinks, and all pages linked from the Ames homepage are required to have a backlink to it; relatively few pages, however, have cross-links for related pages at Ames. For many users, this means that the experience of walking the web at Ames often follows the pattern home-out-back-out-back. Metaphorically, most entry points are like the center point of an asterisk, with one additional line connecting it to a main entry point. What cross-linking does occur relates mainly to resources outside of Ames; it seems to be an assumption of web designers that other resources at Ames are known to the user or are best reached in the home-out-back pattern described earlier.

This backlink pattern is reasonable, if somewhat slow, for users who are following a web path rooted at Ames; for users who enter Ames' space via a non-entry point page linked from outside Ames, the pattern is much more difficult. If, for example, a user follows a link from a page at Stanford on cryogenics to a page describing Pulse-Tube Refrigeration studies at Ames, the only link on the new page will take the user to the divisional homepage for Space sciences. Programs related to cryogenics are several layers below this page and not immediately visible; the search engines available, in contrast, are several layers above. If the Stanford entry point does not contain pointers to the cryogenics research at Ames, the user will likely not find it; even if the Stanford entry point does contain the information, the user is forced back into the home-out-back pattern, with the Stanford page as "home".

Some of those arriving in Ames webspace outside of a main entry point are those using a search engine like lycos, jumpstation, or webcrawler, rather than a subject linked page. >From what we can tell so far, most of these users end up near, but not at the resource they desire. Two basic problems seem to cause this "offset landing" phenomenon. The first is that many of the search engines apparently weight by the number of occurrences of the target word or set of words within a document. This tends to favor subject link pages (pages which attempt to draw together resources grouped around a particular topic) over content pages which directly relate to the topic, unless the target word is repeated frequently in the content page. The second form of offset landing occurs because of users' tendency to describe the type of resource they want as well as the content; for example, many users type search strings like "pictures of the space shuttle", rather than simply using "space shuttle". Since the word "picture" is much more likely to occur in a page describing the photo set or providing a front-end to mission data, the user will end up there rather than at the graphical data itself.3

Offset landing can be a bug or a feature. In many cases, NASA would prefer that the user arrive at a front end page rather than at an image or other data page; it makes it possible to

provide background information once rather than as a wrapper to each page. It can be frustrating for the user, however, and it is only a benefit if the user lands at an entry point that the providers have foreseen.

As the above material shows, one of the main conclusions of our research is likely to be that reliable indexing and searching is possible only when the web design supports a reasonable method of traversal; if it's not browsable, it's probably not searchable either. We also see a few things on a wish list:

1) A way of declaring to robots and spiders noting a page that users should be directed to a different page (an "index page" or a "root page").

2) A way of indicating the type of data which might be returned by a form (for forms which are front-ends to data sets).

3) A way of optionally weighting local resources higher than off-site resources in searches which encompass multiple sites. This would be of especial benefit to those creating documents, as they would be better able to locate local resources for inclusion as cross-links; it might also be of benefit to other users if the aim of a search is to find resources which are geographically or institutionally bound.

4) Dual-method sorting of search results. For example, if a user chose "scoring, host", the search results would be scored and the URLs from a particular host grouped together in output, with the highest scoring host followed by the second scoring host etc.

5) Under the dream, rather than wish, category comes the vision of a search engine that is easily scaled, shares data seamlessly with its peers, stores data in a compact format, caches well, and hogs neither bandwidth nor cpu time.

Dr. Edward Hardie
Network Applications and Information Center
NASA Ames Research Center (Sterling)
Mtn. View, CA
hardie@nasa.gov
1.415.604.0134

Disclaimer: As a consultant, rather than a NASA employee, I do not speak officially for NASA; no part of this document should be taken as official NASA policy..

1The primary additions checked http-referrer and user-agent variables; perl scripts were then used to separate the log entries by host , user-agent, and time. In most cases this produced a pattern which clearly related to a single users' session; in certain cases time factors were ambiguous and the researchers made individual judgments about assigning session boundaries. Only a tiny portion of each day's accesses could be analyzed, so semi-random assignment methods were used to select analysis targets.

2This determination was made primarily from the patterns of revisitation used by those with non-graphical browsers. Other explanations are, of course, possible.

3This problem is made much worse at NASA by the sheer volume of graphical data. Many photo sets or movie archives are stored on multi-player cdrom drives ("Jukeboxes") ; the cdroms involved were originally configured for local access with a specialized player. While access via web browser is possible, the titles of the actual files tend to be short and unhelpful (STS63-L2.gif, for example, is the second gif of a space-shuttle launch which took place in October of 1994, but unless you know the mission number and the coding, the title is completely unhelpful).

Workshop A: Web-wide Indexing/Semantic Header or Cover Page

Chair: Bipin C. Desai, Brian Pinkerton

Kevin Hughes <kevinh@eit.COM>

Abstract
~~~~~~~~

Enterprise Integration Technologies (EIT) deals with a large number of Internet-savvy clients who desire a fast, simple indexing technology that can take advantage of the data they're provided on their World-Wide Web sites. Because of this need, I have developed SWISH - the Simple Web Indexing System for Humans.

SWISH is a program that is both a indexer and a searcher. It builds inverted indices of keywords and stored them in single index files to which HTML gateways can be made. Because SWISH can recognize HTML tags and entities, we can allow it to search internationalized text or search for specific words within particular tags.

SWISH is not meant to be fully-featured - its main strength is that it is extremely easy to use and configure. Already EIT has saved money by giving SWISH to clients rather than having to configure, support, and license WAIS to satisfy the same requirements. Many Web sites do not require a complex, industrial-strength indexing solution - they do require something that is easy to use and HTML-aware, and by keeping with this philosophy, I hope SWISH can fulfill this need.

Outline
~~~~~~~

What is SWISH?

SWISH is a generic indexing and search engine that has as its core philosophy ease of use and HTML awareness. For more information, please see:

http://www.eit.com/software/swish/

http://www.eit.com/cgi-bin/wwwwais

SWISH is also being used at:

http://www.xerox.com/
http://www.city.net/

The Personal Indexer

SWISH is in a class of tools that I call "personal indexers" - these are utilities that allow one to find information that they're looking for at their own site. Glimpse, FFW, and htgrep could be considered to be in this class. What makes all these tools "personal" is mainly the fact that one doesn't necessarily have to be a computer genius to set them up. So ease of use is a definite factor.

"Personality" also comes from the ability to learn from one's preferences and the data itself. If indexing programs work more like signal processors and less like word-grepping beasts, it's possible to make indexing programs both language and topic independent. Look at Architext, for instance. One nice thing about WAIS is that it can narrow one's searches based on feedback. But Architext is proprietary and WAIS is overkill. We need an open, simple solution that doesn't exist yet.

In making such an indexer, one should realize that it would be used to index and search many types of Arabic languages. Thanks to a good deal of international feedback I've been able to make SWISH less language independent - you can define what characters make up words, what certain characteristics of a word are, etc. This feature, it turns out, ends up culling a lot of "garbage" information from index files, shaving about 20% off the index file size (a very rough estimate) or more. This simple filter even seems to work well in extracting "real" words from binary files.

More and more people want to index SGML-like (particularly HTML) structured data. Witness the number of people on comp.infosystems.wais complaining that they can't figure out how to index their Web site. One of the greatest promises of HTML was the idea that one would be able to search and find more easily using structured markup rather than plain old text. So where are all the tools to do this? I believe that the core code is so small, you could include an index/search program with every server, much in the same way that you find an imagemap program everywhere. After all, most Web sites are comprised of about half graphics and other media and only half text. And many Web sites are not large enough to require a full-strength indexer.

Such a well-distributed program would certainly need to communicate with other similar programs, so meta-indexers (like Harvest or GLOSS) and other users could cull them for information. By putting it on the server side (or on a proxy server) it could be contacted via HTTP. I very much intend to add any functionality to SWISH that is needed to make it communicative.

I look forward to hearing ideas from the rest of the folks at the workshop about how we can all share a common language!

-- Kevin Hughes

--
Kevin Hughes * kevinh@eit.com
Enterprise Integration Technologies Webmaster (http://www.eit.com/)
Hypermedia Industrial Designer * Duty now for the future!

Workshop A: Web–wide Indexing/Semantic Header or Cover Page

Chair: Bipin C. Desai, Brian Pinkerton

==========

Defining a High Level Information Gathering System

for the World-Wide Web

David Konopnicki and Oded Shmueli
{konop,oshmu}@cs.Technion.AC.IL
Computer Science Department
Technion, Haifa, 32000, Israel

March 1, 1995

1 Introduction

The WWW can be viewed as a gigantic data repository (mostly read-only). In order to get a piece of information one basically needs to know where the data are located. To facilitate the search, there are certain indexes that are maintained over the WWW (no central control). These indexes are constructed by robots (e.g. WWWW, WebCrawler) that occasionally scan the WWW and construct indexes of interesting keywords.

Still, there is no high level facility for locating, filtering and presenting WWW-held information. In fact, the situation right now is analogous to that of a huge file system, or a document retrieval system, with many useful indexes but without a convenient facility for using this information. One is thus forced to retrieve information manually through browsing and indexes, or write special purpose programs to obtain specific pieces of information.

There is a need to design and construct a high level information gathering and display facility for the WWW.

2 Difficulties in searching the WWW

Currently, access to the WWW is based on navigationally-oriented browsers. This leads to the well-known "lost in cyberspace" phenomenon. Users are confronted with a large, unfamiliar, heterogeneous and constantly changing network. They have no systematic way to obtain information, because of the following reasons:

* There is no reliable road map for the WWW. The WWW is constantly growing and it becomes more and more difficult to locate specific information.

* It is difficult to analyze obtained information. The data found on the WWW is heterogeneous. Some files contain text, while others contain images, sounds or videos. These files are stored in various formats. Therefore, it is currently impossible to verify automatically whether a file satisfies a specific condition (For example, "Find all the images that contain a tree" or "Find all the articles written by A. Einstein").

* In a hypertext environment, the organization of documents conveys information. Nevertheless, it is cumbersome for users to search for information related to the organization of the hypertext.

To address these problems, indexes are built to allow searching for documents, usually based on keyword matching. This approach is useful, but it also suffers from the following problems:

* In indexing, there is, perhaps unavoidable, replication of information.

* Indexes summarize the data, i.e. maintain the portions of the information which are considered important. It is complicated to summarize images, graphics or sound data. So, indexes are most appropriate for text data.

* Indexes do not do well in capturing the hypertext structure of indexed data.

* The existing indexes do not share their information, and they do not provide a common interface.

To summarize, while some help exists, there is no comprehensive facility for information gathering on the WWW.

3 Improvements to the Existing Search Facilities

The WWW supports many file formats and analysis of the data becomes more and more complicated. Search based on keyword matching is not sufficient. An efficient search facility must treat in a uniform way the different file formats used in the WWW, using the information conveyed by the formatting codes.

The WWW is growing, new sites are opening and new users are setting home pages. In the next logical stage,the WWW will grow when users add new items on their servers using the full power of the hypertext environment. Then, the information conveyed by the hypertext structure will grow in importance (for example, the use of the REL argument adds semantic information to the hypertext links). Therefore, a search facility must also consider conditions on the hypertext organization of the searched information.

Searching should let the user concentrate on what he/she wants without worrying about how to get it. The same search facility should be used for communication between search engines and between search engines and indexes, in the development of distributed search algorithms.

Like mechanical robots, software robots do not do well in "real life" environments. The analysis of the environment takes too long for them to be efficient. However, in a "known environment", they may be invaluable. Therefore, there is a need for a formal description language that should be used to make WWW sites "robot-accessible". The protocol for robot exclusion is a step in the right direction.

Maintenance of up-to-date indexes is a complex task. Locating changes in the WWW is very difficult: servers are not stable and new version of existing pages arise constantly. There is a need for servers to manage officially dated maps that will be used to locate changes without having to upload the entire hypertext structure found in a server. Servers should also keep track of home pages (and their rough content) that are located in their "nearby environment". This information should be "robot accessible". This will make robot searches more efficient and less bothersome.

4 Conclusions

An ideal future of the WWW should contain:

* A high level information gathering facility, based on a universal set of concepts which are used to express the information that is searched for.

* Automatic search engines using the information conveyed by the various file formats and the hypertext structure.

* High level management facilities offered by the servers to locate information in an efficient way.

Workshop A: Web–wide Indexing/Semantic Header or Cover Page

Chair: Bipin C. Desai, Brian Pinkerton

Choosing an Indexing Strategy in an Enterprise Environment

Christian Kuhnert, February 27th 1995
(kuhnert@welfa5.elektro.uni-wuppertal.de)
http://welfad.uni-wuppertal.de/people/kuhnert.e.html

Abstract

This paper describes the demands in setting up an indexing system for WWW services within a world-wide operating company. Six systems for description and full text based information recovery, namely Aliweb, Harvest, DIENST, WAIS, FFW and GlimpseHTTP are discussed and compared. After selecting an appropriate solution, some statements about desirable future development are made.

1 Situation

Siemens Nixdorf Informationssysteme AG (SNI) is the largest European manufacturer of Midrange Systems (*NIX Systems). Other important business areas are Mainframes (BS2000) and POS (Point Of Sale) equipment, with a total of roughly 40,000 employees world-wide. Currently the majority of customers reside within Europe. Data exchange between plants and establishments is carried out through a world-wide corporate network. For communications with subsidiaries, partner companies and customers, Internet paths become more and more important. Most of the current data paths are charged on a per volume basis.

2 Targets

The possible internal application areas for WWW are currently evaluated. It is considered to provide Internet connectivity to employees primarily through a WWW interface that integrates most of the services in a user friendly manner. Today there are lots of internal database and information systems, all with their own different user interface. Gateways to these systems are being built. In a special case, migration from a proprietary system to HTTP is considered. Thi system mainly provides product information to sales executives and contains full text index. Representation of this index and its query interface is the main problem herein. Another field of application is the query of internal library catalogues. In parallel WWW services for corporate presence are bein built. Here indexing might be handled in a similar way as with other public servers.

3 Priorities

Currently the limiting factors for any implementation can be seen in the following order:

1. Network cost
2. Storage cost
3. Computation cost

This might not be specific to the SNI environment since in the past years co reductions generally took place in reverse order. From the users perspective the main goals for each service are:

1. Quality of Service (QOS: Availability, Responsiveness)
2. Ease of use (For client as for system administration)
3. Actuality (This has been taken out of QOS)

As you can see there's an inherent conflict between the top points in above listings: QOS is limited by network quality which is almost proportional to cost. Reduced to indexing and retrieval the main questions are:

Q1: Index distribution cost versus query transfer cost
Q2: Local computation versus WAN access

4 Review of Common Methods for WWW Server Index Generation

Besides robot based gathering from information sources (like Lycos, WWWW, RB Spider and WebCrawler) the most frequently used system with server side supp is ALIWEB[1]. A new and more general approach is followed by the HARVEST[2] system. In short HARVEST trades network load for storage cost – a reasonabl choice regarding the given priorities. The DIENST[3] Protocol focuses on distribution of academic papers. It relies upon bibliographic descriptions i RFC-1357 format and distributes queries to multiple locally maintained databases. None of these systems support full text indexing (Harvest could, would loose its advantages; DIENST actually does, but relies on WAIS[4] for implementation), therefore WAIS must be considered. Finally there are specialised full text indexers that support HTML: FFW[5] and GlimpseHTTP[6].

4.1 Description Based Indexers

ALIWEB

Aliweb is based on description files that are collected at regular intervals then combined into a searchable index. It is the information provider's responsibility to compile and update this description file which follows a common standard. This can be done manually or using some information extract tool. Aliweb currently depends on one master server for gathering data. The

index then is mirrored.

HARVEST

Harvest essentially can be broken up into two main components called "Gatherer" and "Broker". The Gatherer extracts object descriptions from files of known type (besides HTML this includes binary, some graphics formats, etc.) and exports them via its TCP port. A proprietary, structured format called SOIF (Summary Object Interchange Format) has been defined to exchange these descriptions. The Broker connects with one ore more Gatherers (or other Brokers) to collect this data (optionally compressed with gzip) to build an index. It then accepts query requests by listening to an own TCP port. Structured queries using Boolean expressions and fielded search are provided.

DIENST

DIENST (which stands for Distributed Interactive Extensible Network Server for Techreports) provides an HTTP based protocol for structured search in distributed databases and object oriented document retrieval. It creates an index from bibliographic description files in RFC-1357 format that can be searched on each server. Similar to WAIS there is a master index of servers that can be used to forward queries to the appropriate sites. Documents can then be retrieved in a variety of formats. Recently also fulltext search is supported using the SMART search engine (which provides a look and feel that is similar to WAIS) or WAIS itself.

4.2 Full Text Indexers

WAIS

The WAIS (Wide Area Information Servers) system allows for full text search in a variety of databases, distributed on the network. A single directory of servers lists available WAIS indexes. Users can select appropriate servers and pose a query to them. Found items will be presented with a relevance rating depending on the number of occurrences of the keywords from the query in the document. A major drawback of WAIS concerning data distribution is that for a query to be answered not only the index but also the underlying database must be accessed. Therefore data and index information are kept in the same location. Also WAIS indexes are around the size of indexed data or even larger to provide fast search. There are different implementations of WAIS available, some of them supporting Boolean expressions and date search.

FFW

FFW (Freetext Search for the Web) is a fulltext indexing system that focuses on HTML documents. One of the advantages over WAIS in its application is that it generates a "self-contained" index: Only the index data is needed to answer a query. It provides a means for merging large indexes from existing smaller ones and to distribute queries amongst indexes which are scaled around 30% of dataset size. Only simple queries (but including expression grammar, word truncation and date search) are supported.

GlimpseHTTP

Glimpse is usually the underlying indexing mechanism for Harvest object descriptions in a Broker. It can also be used standalone with some small extensions to provide fulltext search on HTML documents. As with WAIS, Glimp needs to access the indexed data to satisfy a query, but allows index size t reduced to around 7% of data size by trading access speed. Glimpse supports wider set of queries, including spelling errors and regular expression match

5 Evaluation

The presented description based indexers are very different in scope and implementation. While Aliweb has the appearance of being an ad hoc solution the resource location problem, the others are more designed and allow for hierarchical index arrangement (Harvest) or query distribution (DIENST), eas expansion (both) and abstraction from files (DIENST).

Concerning full text retrieval, WAIS is the most common and general syste and GlimpseHTTP are more lightweight solutions which focus on WWW servers. T both have their individual advantages (e.g. FFW dealing with the full ECMA Latin-1 character set and providing a self-contained index; GlimpseHTTP bein very unpretending about disk space). They lack a mechanism to build larger indexes from existing ones as WAIS (virtually) does.

	explicit data description needed	indexed objects	phys. location of index	levels of hierarchy	query executio
Aliweb	yes	sites, files, services	centralised	1	on master or mirror
Harvest	no (generated by essence)	files	arbitrary	n	on master or replic
DIENST	yes	documents	distributed on sites	2	distribut
WAIS	no	files	distributed on data sites	2	distribut
FFW	no	HTML files	arbitrary	1	on index
Glimpse HTTP	no	HTML files	on data site	1	local

The table shows some key characteristics of the six indexing tools discu this paper.

Since the scope is HTTP retrieval within one enterprise, Aliweb must be d WAIS would be a preferable choice if it was already used as a retrieval syst within the company. Since it isn't, the other, more WWW oriented packages ca considered. DIENST contains some very good ideas about handling different da formats of the same document, but currently is limited to documents with bibliographic descriptions available. It's a very promising approach for on- library services and presents a friendly user interface.

For the final decision, a look on the priority list might help: As we hav network load is the major concern. It is difficult to estimate that paramete for the different solutions. Statements like "can reduce [...] network traff

by a factor of 59" (from [2]) treated with care. A system that is flexible enough to allow a decision on distribution policy while being used would be preferable. This makes Harvest the most promising solution. After a gatherer is running locally for every resource, brokers can be set up at any location.

For full text indexes, Harvest produces too much overhead: In its present implementation a (compressed) SOIF object containing the full document must be stored by the gatherer, be transferred to a broker and get indexed. The index then could be replicated. The same functionality could be achieved with FFW, using standard mirroring for the index. GlimpseHTTP does not meet this requirement as its index is not self-contained. Only Harvest and FFW make it possible to give a "flexible response" to Questions Q1 and Q2 for the desired application.

6 Application

For indexing the internal and external WWW server contents Harvest is used. With Version 1.0 several major bugs showed up, that disappeared when upgrading to V1.1 these days. Administration of the Harvest system, beeing very complicated wit V1.0, is also more straightforward in the new version. For full text index generation on the product database FFW will be used.

7 The Future

For further development, it would be desirable to integrate index generation with revision control. At present, revision control is the lacking element in providing WWW documents. It should be integrated within the server - as proposed by the HTTP protocol - as a handler for the PUT and DELETE methods. When this step has been taken, forming a database from an ugly heap of files, some of the former indexing problems will have disappeared.

When the versioning system detects a change, it could initiate an incremental index update, thus removing the need to process the whole database at regular intervals. The changes in the index then can be propagated to registered sites as delta information, minimising network load. This is comparable to the transition from procedural to event driven programming.

Instead of running batch jobs in the night (when is "night" in a world-wide web anyway?) to update an index, recently developed on-line index construction algorithms must be used[7].

This form of version tracking and change propagation will also help to solve the notification problem as described in [8].

References

[1] Koster, Martijn: Welcome to ALIWEB. On-line document (http://web.nexor.co.uk/public/aliweb/aliweb.html)Nexor, UK, February 1995

[2] Hardy, Darren R. and Michael F. Schwartz: Harvest User's Manual, Version 1.1. Technical Report CU-CS-743-94, University of Colorado at Boulder, February 1995.

[3] Davis, James R. and Carl Lagoze: A protocol and server for a distributed digital technical report library. (http://cs- tr.cs.cornell.edu/Server/TR/CORNELLCS:TR94-1418) Cornell University, April 1994.

[4] Marshall, Peter: WAIS: The Wide Area Information Server or Anonymous What???. (ftp://ftp.wais.com/pub/wais-doc/UWO-wais-paper.ps) University of Western Ontario, June 1992.

[5] Hfjeld, Brd: FFW - Freetext search for the Web. On-line document (http://www.nta.no/produkter/ffw/ffw.html), Telenor Research, Norway February 1995.

[6] Klark, Paul: GLIMPSE, A tool to search entire file systems. On-line document (http://glimpse.cs.arizona.edu:1994/) University of Arizona February 1995.

[7] Srinivasan, V and Michael J. Carey: On-line Index Construction Algorithms. (http://www.cs.wisc.edu/TR/UWMADISONCS:CS-TR-91-1008) University of Wisconsin, February 1991.

[8] Notification of new material. On-line document (http://info.cern.ch/hypertext/WWW/DesignIssues/Notification.html) CERN, October 1993.

Workshop A: Web–wide Indexing/Semantic Header or Cover Page

Chair: Bipin C. Desai, Brian Pinkerton

WWW Indexing Workshop Proposal for Participation

Nancy B. Lehrer
ISX Corporation
4353 Park Terrace Drive
Westlake Village, CA 91360

email: nlehrer@isx.com
URL: http://isx.com/~nlehrer/
phone: 818-706-2020

VIeW (Virtual Information Web) Project at ISX

ISX is currently undertaking several projects aimed at making Web information more accessible. Our main domain focus has been ARPA project information webs, but we will soon be branching out into the education domain to support the sharing of K-12 educational curricula. This work is currently supported by the ARPA Intelligent Information Integration (I3) Initiative.* The project is described in detail at the URL http://isx.com/~nlehrer/I3/view/view.html. You can also see the humble beginnings of a VIeW tool kit from the Intelligent Integration of Information (I3) Initiative home page at http://isx.com/pub/I3. Go to the I3 Project Lists link.

The VIeW (Virtual Information Web) Project

The purpose of the Virtual Information Web (VIeW) project is to build tools which aid in the dissemination of ARPA project information using the World Wide Web.

The World Wide Web is primarily a passive environment. When a person or organization has some information to share, they create a web to describe their ideas, products, research, and hope that others take a look and get interested. There is a lot of information out there and there are a lot of people looking for information, yet there are few tools for querying information beyond indexed searches. Additionally, the synthesis of information available on the web is a difficult task due to widely varying presentation styles and information organization.

The Virtual Information Web Project attempts to address the some of the issues by abstracting project webs into structured information which can be viewed and queried in multiple ways. Additionally, the VIeW project has experimented with distributed database search over the web. Unfortunately, this approach is currently sidelined due to the large amount of support it requires at each individual site. ARPA sponsored research and technology initiatives will provide the initial domain and requirements for this project.

Goals

The goal of the VIeW Project is to enable a virtual initiative information web where each ARPA contractor maintains their portion of the web locally in a locally preferred style, yet the information is made available on an aggregate level in a globally consistent style. The general approach is to have ARPA initiative participants develop local information webs including:

- Project Overviews and Technical Summaries
- Address books, Calendars, Bibliographies
- Administrative data

The goals of the VIeW project are to enable project information abstraction to support multiple project information views, project queries, and aggregate initiative information.

Virtual Information Web Challenges

The Virtual Information Web project faces the following challenges for acceptance.

Near zero-energy for technologists
The VIeW solution must support a near zero-energy approach for the technologists.

Perceived Benefit for Technologists
The benefit from ARPA point of view is clear. There must also be a perceived benefit for a Technologist to use the VIeW tools. VIeW must illuminate the increased visibility of their project to both their customer and other technologists.

Security
VIeW must be secure. Sensitive information must be guarded. VIeW tools must not open security holes in the

host system.

Usability
VIeW must be easy to use and must not require learning a large new set of tools. Optimally, VIeW would support using tools the user is currently familiar with.

VIeW Approach:

The success of the VIeW will depend upon finding the specific project information and project description structure underlying the generally unstructured environment of the Web.

To enable structure discovery, the VIeW approach asks that technologists re-visit their webs and add annotations which identify the project information such as project motivation, overview, goal statements, recent accomplishments, and unique technology contributions. A web crawler, the VIeW Maker, will then be responsible for parsing web documents and creating a structured database on the VIeW server. Additionally, the VIeW Maker will index project webs using one of the currently available text indexing schemes to add keyword search capabilities to the webs.

*This work funded by the Intelligent Information Integration Initiative of the Advanced Research Projects Agency, SISTO F33615-94-1556. Mr. Dave Gunning ARPA Program Manager.

Workshop A: Web-wide Indexing/Semantic Header or Cover Page

Chair: Bipin C. Desai, Brian Pinkerton

Michael L. Mauldin 1 March 95
Carnegie Mellon University
Pittsburgh, PA 15213-3890
fuzzy@cmu.edu
http://fuzine.mt.cs.cmu.edu/mlm/

The Lycos (tm) Catalog of the Internet is a collection of rich abstracts of texts available on the World Wide Web. These abstracts are automatically generated by the Lycos robot, which continually scans the Web looking for new documents and checking older documents for changes. As each document is read, a full abstract is produced. Further, any document referenced by a hyper-link is given at least a short description.

Although the collection process is primarily automatic, we do provide a URL registration service to allow users and authors to include documents in the catalog; further, we provide mechanisms for authors to remove their documents from the catalog or keep them out in the first place.

Carnegie Mellon University also provides a query retrieval service against this catalog, serving over a half a million retrievals to over 140,000 users a week. We are committed to providing ubiquitous global access to this catalog retrieval service, and are working on licensing arrangements to ensure its continued availability.

Carnegie Mellon is also committed to scaling up the collection process to the entire World Wide Web and maintaining this collection current to within a month. We currently have downloaded 10% of the estimated web within the previous 3 months, and have identified two thirds of the documents by URL. We are scaling up incrementally, continuing to provide a world wide retrieval service against the entire catalog. Since November, we have added new computers to this effort at the rate of one every two weeks.

We believe common or compatible data and index formats as critical issues for this workshop. A minimum level of interoperability would cover the data format for document and abstract representation. More difficult is to define a common format for indexes (inverted files), since these are often specialized and in some cases proprietary. For example, Lycos/Pursuit uses word position information not stored in WAIS indexes. Some WWW search engines match regular expressions against the whole document collection and do not need inverted files.

Workshop A: Web–wide Indexing/Semantic Header or Cover Page

Chair: Bipin C. Desai, Brian Pinkerton

Intelligent Web Servers as as Approach to the Web Indexing Problem

James Mayfield and Charles Nicholas
Computer Science Department
University of Maryland Baltimore County (UMBC)

Like the rest of the Web community, we have watched the Web explode in size and popularity over the last several months. We have also seen the emergence of robot-based indexing tools, such as Lycos and the World-Wide Web Worm. In our research, funded in part by the US Department of Defense, we are designing what we call "Intelligent Web Servers". These intelligent web servers will address some of the problems of scalability and currency that adversely affect the robot-based approaches.

The Web servers we have in mind are intelligent in the following sense: (1) they possess metadata, i.e. knowledge about the information they can provide, and (2) they can communicate with each other in order to better handle user requests.

With respect to the metadata the servers will have, we are using two approaches, both grounded in our earlier work in this area. In the design and construction of the SNITCH system we devoted considerable effort to the use of semantic nets as a means for constructing hypertext links, and obviously for such automatic linking to be effective, this semantic net must be descriptive of the underlying corpus.

In other work related to metadata, we use statistical properties of the text itself, namely the distribution of n-grams, again to construct hypertext links. The TELLTALE system is based on the assumption that documents with similar n-gram profiles are indeed similar in content. We are now involved in work to upgrade the TELLTALE in terms of performance and capacity, and adapting TELLTALE to the WWW is an important aspect of this work.

In addition to having metadata, our intelligent web servers will be able to communicate with each other. Right now, Web servers don't cooperate with each other in any meaningful sense, with the possible exception of proxy servers and caching. We propose that if, for example, a Web server is asked for an HTML file that it doesn't have, it asks its peer servers if they happen to have the requested file. This cooperation will, for example, help shield outside users from the possible ill-effects of migration of files from one server to another server on the same LAN.

These intelligent web servers will communicate with each other using KQML, which stands for Knowledge Query Manipulation Language. The syntax of KQML is well-defined, and a few more-or-less stable implementations are available. Development of the syntax and semantics of KQML continues at UMBC and a number of other places.

At the moment, our work is very much in progress. We have only recently started to incorporate KQML functionality into a Web server. As yet we have no experimental evidence to offer. However, once we have a small community of these intelligent Web servers, we plan to endow each of these with some data (and metadata) to manage. We will then run experiments on how well these servers deal with (1) information requests that involve possibly outdated or otherwise broken URLs, and (2) information requests of a more nebulous nature, such as "Show me more HTML documents that resemble document X."

Charles Nicholas Computer Science Department, UMBC
410-455-2594, -3969 fax http://ruff.cs.umbc.edu:1080/

Workshop A: Web–wide Indexing/Semantic Header or Cover Page

Chair: Bipin C. Desai, Brian Pinkerton

Leon Shklar

Over the last few years, there has been a proliferation of different indexing technologies. There has also been a proliferation of applications and information management systems that handle specific types of data (text, images, structured, etc.). We believe that it is unrealistic to expect that all massive amounts of existing heterogeneous data will ever get converted to a single format, that everyone will use a single indexing technology, or even that all different retrieval engines will use the same indexing information:

1. It would be prohibitively expensive to convert all the existing data and all the existing indexing information into single representations. It would be almost just as bad to have to support backward compatibility of existing tools and applications.

2. Given the diversity of both the existing information and the retrieval objectives, any single representation of indexing information would most likely be redundant to the point of being impractical.

3. New data formats and representations, as well as new indexing technologies will continue to emerge.

We believe that the same approach should be adopted in dealing with both the legacy data and the legacy indexing structures. We are developing a declarative language to support object encapsulation of both data and indices [1]. We intend to treat each indexing technology as a black box and to find meaningful ways of combining results of querying different heterogeneous indices.

Of course, the weights of selections computed when running a query against a particular index generally make sense only in the context of that index. We see the major challenge in using statistical and machine learning methods to scale the weights computed for different (possibly heterogeneous) indices against each other.

Our statistics-based approach is to view a query against several different indices as the problem of predicting an unknown outcome based on the observe values of multiple heterogeneous predictors. The methods of machine learning provide the ability to perform similar extrapolation from training sets of queries. The general goal of inductive learning is to generalize from labele data and form rules for accurately labeling future, unlabeled data. In this case, the results of learning will be procedures for user-specific filtering of results based on the user's history of data access (from examples of appropriate and inappropriate retrievals).

[1] L. Shklar, K. Shah, and C. Basu, "Putting Legacy Data on the Web: A Repository Definition Language", To be presented at the WWW'95, Darmstadt, Germany.