Design and Development of a Network-Based Electronic Library [*]

Ray R. Larson
Associate Professor

School of Library and Information Studies
University of California, Berkeley
Berkeley, CA 94720
ray@sherlock.berkeley.edu

Abstract

Among the proposed innovations in the Clinton Administration's plans to develop a *National Information Infrastructure* is the creation of, and support for, *digital* or *electronic libraries* to store and provide access to the vast amounts of information expected to made available over the "information superhighway". Although the exact nature and future architecture of such libraries is still a matter for experimentation (and debate), there are several pioneering efforts underway to establish electronic libraries and to provide access to them.

This paper describes one such effort underway at the University of California at Berkeley. In collaboration with four other universities we are developing interoperable electronic library servers containing the Computer Science technical reports for each participant and making them available over the internet using standard protocols.

1 Introduction

This paper describes the design and development of an *Electronic Library* for Computer Science technical report literature currently under development at the University of California at Berkeley. The planned library is the result of a collaborative research project involving UC Berkeley, Stanford, Cornell, MIT, and Carnegie-Mellon University sponsored by CNRI (Corporation for National Research Initiatives) and ARPA. This project has the initial goal of putting the full text and scanned page images of all Computer Science technical reports for each participant into locally mounted network-based servers with a common retrieval protocol. This paper concentrates on the contributions and development efforts at UC Berkeley in this effort to develop an electronic library and does not necessarily reflect the views of CNRI or the other participants. Two areas of the design and development for the Computer Science Technical Report (CSTR) electronic library

[*] The work described in this paper was sponsored by the Advanced Research Projects Agency (ARPA) of the Department of Defense under Grant No. MDA972-92-J-1029 with the Corporation for National Research Initiatives (CNRI). The views and conclusions contained in this paper are those of the author and should not be interpreted as necessarily representing the official policies or endorsement, either expressed or implied, of ARPA, the U.S. Government or CNRI. A version of this technical report was published in *Navigating the Networks: Proceedings of the ASIS Mid-Year Meeting* Portland, Oregon, May 21-25, 1994

will be discussed. The first is the conceptual model of the electronic library that is driving the development efforts, and how that conception is being turned into a working electronic library. The second area is a discussion of the longer term basic research issues that will need to be addressed as electronic libraries proliferate in a global network environment. The Berkeley group working on the CSTR project includes faculty and graduate students in the School of Library and Information Studies and the Computer Science Department[1].

The following sections of this paper will the describe the view of electronic libraries that has been driving the work at Berkeley on the CSTR project and how that view translates into an architecture for future electronic libraries. This will be followed by a brief discussion of the current and planned architecture of the Berkeley server. The indexing and retrieval methods and the database management system technology underlying the Berkeley server will be examined. Finally, the paper will describe some of the research issues in the design and development of electronic libraries and future research agenda of the CSTR project and other electronic library projects at Berkeley.

2 Defining the Electronic Library

The design and development of the Berkeley CSTR electronic library has been based on a particular vision of electronic libraries and how they will develop and proliferate in the future. This vision assumes that there will be many sites that will want to contribute material to and access material in "The Library". This library will be a global *virtual library*, the traditional collection model of gathering all information possible into one place is no longer tenable, or desirable. We envision a vast population of users scattered around the globe who are able to access, easily and conveniently, the complete contents of thousands of large and small repositories containing texts, images, sound recordings, videos, maps, scientific and business data, as well as hypermedia combinations of these elements. The library must, therefore, be a network-based distributed system with local servers responsible for maintaining individual collections of digital *documents* ranging from sets of electronic texts to video-on-demand services. In effect, this virtual library will actually consist of a set of *publishers* of electronic information and a set of *consumers* distributed across the network.

The "glue" that holds together this distributed library will be conformance to a set of standards for document description and representation, and a set of communication protocols. We believe that the use of multiple standards for both document description and representation and for communication are inevitable in the near term. For eventual standardization of document description and representation, we believe that SGML will provide the basis for text and compound document architectures (supported by additional standards for image, video, and compound documents). For communications we believe that the standard protocol, for low-level query/response and document delivery, will be some extended version of the ANSI Z39.50 information retrieval protocol. Currently we are supporting multiple communications protocols for access to the contents of the CSTR database (including the POSTGRES libpq interface, the World-Wide-Web's HTTP, Gopher, and FTP). Eventually a higher-level protocol, such as CNRI's KIS (Knowbot information server) may be used as well.

[1] Faculty participants are Profs. Robert Wilensky (PI) and Michael Stonebraker of EECS, Michael Buckland and the author of SLIS, also participating is Clifford Lynch of the UC Office of the President, Division of Library Automation. Use of "We" in this paper represents the author's interpretation of the combined view of the faculty investigators. This paper is largely based on the initial Berkeley proposal to CNRI(Wilensky *et al.* 1992).

This standards-based model allows individual information providers (or *publishers*) to experiment with different implementation schemes while allowing the system to scale and preserve interoperability. That is, the use of open standards and the distributed client/server model will permit production servers (even commercial servers) to operate in parallel with experimental research-oriented servers, and will permit both to be accessed from any client. Materials in the Electronic Library may accessed via any protocol-compatible client, and provided by any protocol-compatible server.

In general, we agree with the 9 "principles" for electronic libraries discussed by Fox (Fox *et al.* 1993):

1. *Declarative representations of documents should be used.* Although most of our collection is now in page image and Postscript form, OCRed text is available, and we are hoping to develop SGML markup version though automatic parsing of image and text. However, at least for the present and near future, multiple representations of documents should be supported and available.

2. *Document Components should be represented using natural forms, namely* objects *that can be manipulated by users familiar with those objects.* Currently familiar objects (lists of authors and titles, page images, etc.) are maintained in the database and presented to the user.

3. *Links should be recorded, preserved, organized and generalized.* The database schema developed for the POSTGRES database supporting the CSTR library provides for any object or item in the database to be linked to any other. Each object also has a unique object ID that can be referenced to provide such linkages. For linkages to external objects and databases, we plan support in the CSTR server for URLs and URNs (*Uniform Resource Locators* and *Uniform Resource Names*, the former have become more well know as the hypertext addresses used in the World-Wide-Web protocol.

4. *There should be separation between the digital library and the user interfaces to it.* We strongly support this principle, the entire model for the electronic library is oriented towards client/server operation, with interfaces left entirely to the client-side implementation. At the present time we have a number of separate interfaces that provide access to the same underlying database.

5. *Searching should make use of advanced retrieval methods.* We have been extending the POSTGRES post-relational DBMS to incorporate advanced indexing and retrieval techniques into the system. The current version uses a probabilistic retrieval algorithm that ranks retrieved documents in order of their estimated probability of relevance w.r.t. the user's query. We are planning to include additional database support for access methods that will provide both effective and efficient indexing and retrieval in support of advanced IR methods.

6. *Open systems that include the use, and where (some of) the functions of librarians are carried out by the computer, must be developed.* Work is underway to develop an *Automated Librarian* using natural language processing (NLP) techniques that will locate information relevant to a user's request by understanding a potentially relevant text and the user's request at as deep a level as is necessary and possible, and by applying its knowledge about where information of various sorts is likely to be located.

7. *Task-oriented access to electronic archives must be supported.* Since the electronic library is being developed in conjunction with the Sequoia 2000 project, the database includes much

more information than just the contents of the CS technical reports. The "Sequoia side" of the project seeks to integrate the functions of the electronic library with the task-oriented data storage and retrieval needs of the scientists using the database.

8. *A user-centered development approach should be adopted.* The project has been largely driven by the needs of the population that will be using the electronic library. We plan to incorporate full-scale user studies and analysis as the project continues.

9. *Users should work with objects at the right level of generality.* User interactions and needs with regards to the electronic library and the objects in it and in the database are not yet well understood, but the library should be flexible enough to permit each user to deal with the information in the forms most appropriate to his or her needs.

3 The Berkeley CSTR Library Architecture

The Berkeley group views the electronic library as a modern database application, and we are building it on top of the POSTGRES next-generation DBMS(Stonebraker & Kemnitz 1991). The development effort at Berkeley is being carried on in parallel with Sequoia 2000, a research project that is developing a very-large-scale multimedia object server, with a focus on managing scientific data on global climate change.

We expect that the electronic library will be a useful tool for Sequoia researchers and that documents useful to them will be made available along with the Computer Science technical Reports. We also expect that Sequoia will provide us with a variety of alternative types of data (including images, video, etc.) that will enrich the universe of data types available for our exploration and use in the electronic library. There has been much synergism between the two projects. The Sequoia project has provided a base of workstation hardware and large-scale data storage devices that are being shared to provide computing and storage resources for the electronic library project. We are also sharing some of the software that has been developed for each project. The CSTR project has modelled its overall system architecture to support the electronic library of technical reports in part on the DBMS-centric architecture developed for the Sequoia project, and in part on the current *de facto* standard applications for network search and retrieval of information. Figure 1 shows the overall architecture of the CSTR electronic library system in its current configuration.

This architecture can be considered as set of functional layers mediating between the user of the system and the stored data. These functional layers are the Application or User Interface layer, the Network Protocol layer, the Data Management layer, the Filesystem layer, and the Device layer. The following sections will examine each of these layers, from the Device Layer up through the User Interface Layer, for the current CSTR system.

3.1 The Device Layer

The contents of the CSTR electronic library are currently stored on both magnetic disk (primarily for indexes and programs) and on an optical disk jukebox containing the full text and page images of the technical reports. Because most of the technical reports were not available in a machine-readable form, the database is made up primarily of scanned page images, and the output of OCR run on those page images. Some of the more recent reports are available in PostScript or in some

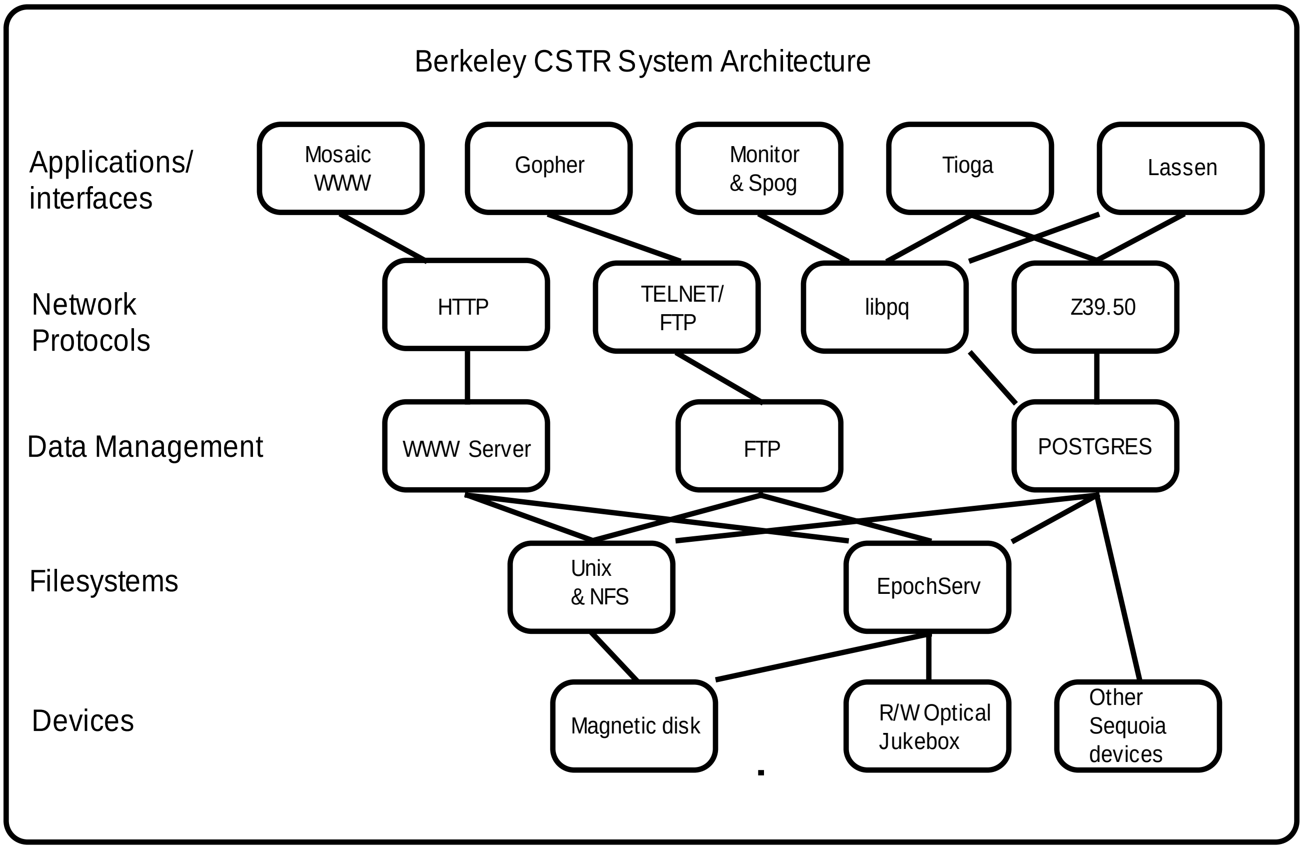

Figure 1: The Berkeley CSTR Server: Overall Architecture

mark-up language (such as troff or TeX). The database was created by scanning each report using a PC and commercial OCR software, and creating a simple bibliographic record for each item. These bibliographic records use a format developed for exchange of CSTR information via email(Cohen 1992).

3.2 The Filesystem Layer

In addition to the conventional UNIX and NFS file systems supporting the magnetic disks for the electronic library, we are using the EpochServ filesystem software to support access to the optical disk jukebox. For all intents and purposes this software hides the fact that data may be written on multiple platters in the optical jukebox. Directory information is maintained on magnetic disk for faster search and access by name, and some data caching is provided for frequently used data sets. In the current configuration, the library of technical reports appears as a normal UNIX hierarchical file system with separate directories for each report and individual TIFF image files for each page in the report, or one or more PostScript files.

3.3 The Data Management Layer

At the present time the electronic library supports a number of data management and access tools (discussed below) to provide access to the database. Some of these tools, such as Gopher and simple FTP rely on the hierarchic structuring of the page image and text files to provide access. The World-Wide-Web server provides access to the technical report via indexes maintained by a set of perl scripts and “c” programs.

However, the primary electronic library server under development at Berkeley is a POSTGRES-based system that can communicate using the Z39.50 protocol. POSTGRES is a next-generation DBMS that supports user-defined abstract data types, user-defined functions, a rules system, and many of the features of Object-Oriented DBMS (including inheritance and *methods* via functions in both the query language and conventional programming languages). Details on the use of POSTGRES for indexing and retrieval in the CSTR electronic library are provided in the next section.

Further extensions to the POSTGRES system are underway. Of particular importance is the development of a Z39.50 compliant interface to the POSTGRES system. Our server will function by mapping requests in the standard Z39.50 protocol into queries the POSTGRES query language. We are also in the planning stage of development to include additional database support for a new class of DBMS access methods that will provide both effective and efficient indexing and retrieval in support of advanced IR methods.

3.4 The Network Protocol Layer

With the variety of database servers and user interface clients available for the electronic library, there are a corresponding number of network protocols supporting those clients. The primary protocols supported are the standard TCP/IP suite of telnet and FTP protocols. The conventional FTP and gopher clients communicate with the electronic library using the FTP protocol, the WWW and mosaic clients use the HTTP protocol.

The current POSTGRES interfaces, such as the POSTGRES ”monitor” or ”spog” and the Tcl/Tk browser implementations communicate with the POSTGRES DBMS using the POSTGRES libpq network interface. The libpq interface provides a low-level socket connection and transfers queries and data between the POSTGRES *backend* database server and the *frontend* clients using a set of “c” library routines.

As mentioned above, we are building support for the Z39.50 information retrieval communication protocol into the POSTGRES database system. Support for Z39.50 is already built into the *Lassen* client software described below. This Z39.50 client software has been tested using existing Z39.50 servers, such as the UC MELVYL system. Providing Z39.50 access to the database opens up the electronic library to any client program that uses this protocol. We believe that the Z39.50 protocol, as an official ANSI standard, has the potential to become the standard protocol for search and retrieval from electronic libraries. The ongoing enhancements to the Z39.50 standard appear to offer all of the capabilities of the current *de facto* standards like the HTTP protocol used in Mosaic.

3.5 The Application or User Interface Layer

The CSTR database is currently made available via a number of interfaces over the internet. These interfaces include any WWW or Mosaic client, gophers, FTP, and some POSTGRES interfaces. For those interested the WWW URL address is *http://tr-ftp.CS.Berkeley.EDU/*. FTP users can access the technical report database using *anonymous@tr-ftp.CS.Berkeley.EDU.*

We have also produced a graphical user interface to the electronic library. This interface has the capability of communicating with information servers using the Z39.50 communication protocol, and also to interact with POSTGRES using the libpq interface. The preliminary version of this user interface for the electronic library of technical reports is based on the Lassen Text Browser interface developed for the Sequoia 2000 project. This interface was designed to separate user interactions from the underlying search and retrieval engine, and to operate in a distributed environment where these components may be running on separate machines of a local or wide-area network. The interactive elements of this interface are based on the Tcl language and Tk toolkit interpreters(Ousterhout 1990; Ousterhout 1991), and thus provide a high degree of flexibility in laying out and modifying the interface elements. Because the interface is based on interpreted code, it can be changed virtually "on-the-fly" for testing and improvements, but still provides remarkably fast and efficient screen manipulation.

In addition, the Berkeley group is developing a visually-oriented browser for the electronic library called "Tioga"(Stonebraker *et al.* 1993). This interface, developed for the Sequoia Project, provides a set of visualization tools for interacting with the database in new ways. In Tioga, the user links together boxes and arrows representing database or processing functions and data flows (rather like the paradigm used in many visual programming languages, and in such systems as Khoros). The builtin tools of Tioga permit the user to present graphically the results of these operations on the data. The user might, for example, generate a plot of term frequency for some selected terms vs. the date that the report was generated. The *ad hoc* sorts of data analysis available in Tioga make it a very powerful tool for a variety of context. We plan to give this browser the capability to generate requests in the Z39.50 standard protocol, and to add a variety of text analysis tools to its set of of "ingredients"

4 Indexing and Retrieval in the CSTR Electronic Library

As noted above, the primary engine for information storage and retrieval in the CSTR library is the POSTGRES next-generation DBMS. POSTGRES supports user-defined abstract data types, user-defined functions, a rules system, and many of the features of Object-Oriented DBMS (including inheritance and *methods* via functions in both the query language and conventional programming languages) that make it suitable to extensions intended to support advanced information retrieval in the DBMS.

We have been using these features of POSTGRES to provide advanced indexing and retrieval techniques for the electronic library system. This section will describe the processes used in the current version of the *Lassen* indexing and retrieval system, and also provide a glimpse of some future plans for further support of IR methods in the DBMS.

4.1 Indexing

The current indexing method runs as a *daemon* that is invoked whenever a new bibliographic record or full-text document is appended to the database. There are a number of POSTGRES database relations (or *classes* in POSTGRES terminology that are used to support the indexing and retrieval process. These classes and their logical linkages are shown in Figure 2. The *wn_index* class contains the complete WordNet dictionary. It provides the normalizing basis for terms used in indexing text elements of the database, that is, all terms extracted from data elements in the database are converted to the word form used in this class. All other references to terms in the indexing process are actually references to the unique IDs assigned to words in this class. The *wn_index* dictionary contains both individual words and common phrases, although in the current implementation only single words are used for indexing purposes.

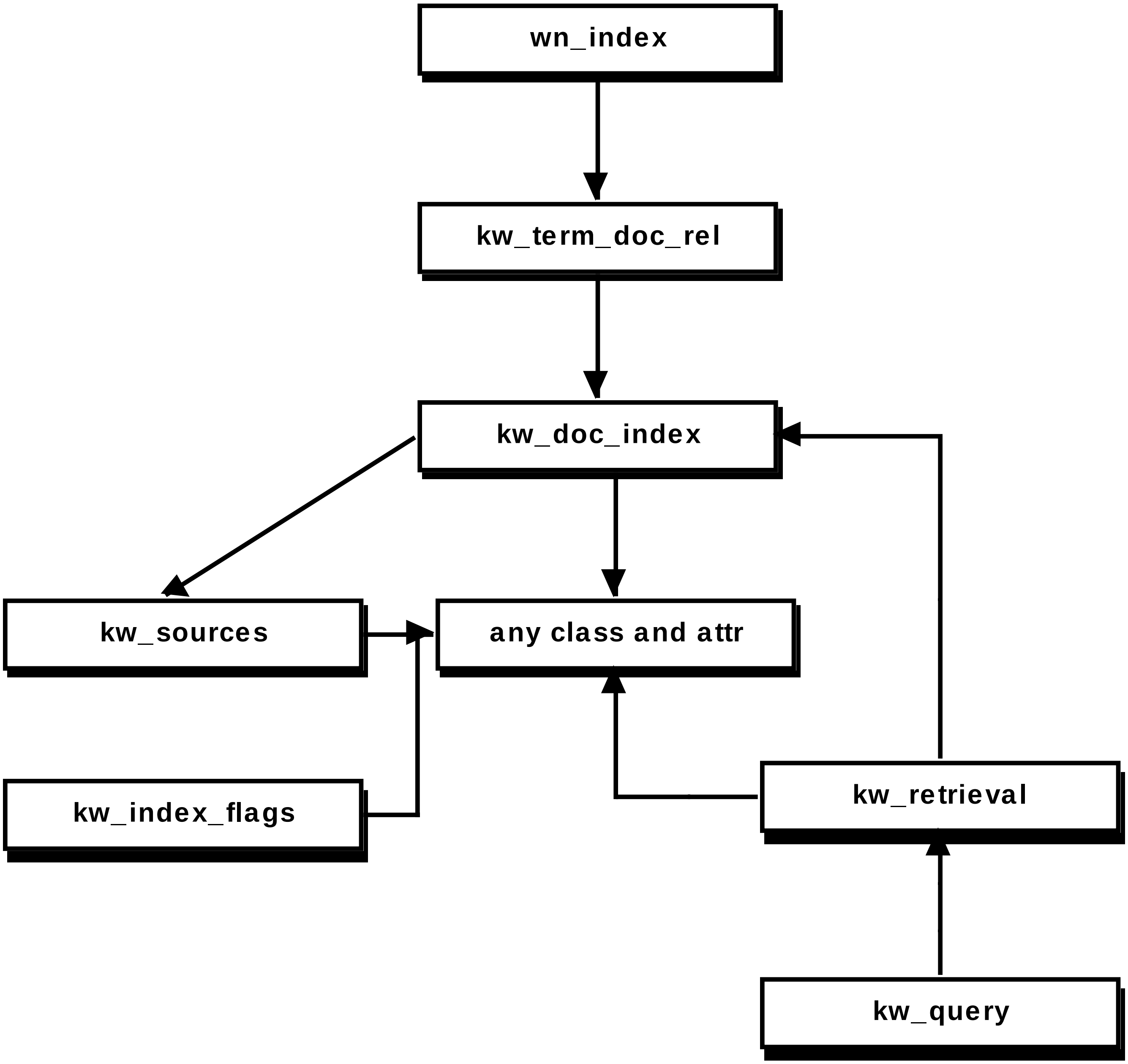

Figure 2: The Lassen POSTGRES Classes for Indexing and Their Linkages

The *kw_term_doc_rel* class provides a linkage between a particular item or data element (both considered *documents*) and a particular term from the *wn_index* class. The raw frequency of occurence of the term in the document is included in the *kw_term_doc_rel* tuple. The *kw_doc_index* class stores information on individual *documents* in the database, including a unique *document ID*, where the document is located (what class, attribute and tuple contain it), and whether it is simple attribute or a large object (with effectively unlimited size). Additional statistical information (such as the number of unique terms found in the document) is maintained in the *kw_doc_index* class. The *kw_sources* class contains information on the classes and attributes indexed at the class level, as well as statistics such as the number of items indexed from any given class. The other classes shown in Figure 2 are involved in the mechanics of indexing and retrieval.

The POSTGRES rules system is used to both ensure that the elements of the bibliographic records are stored in their appropriate normalized form, and to trigger the indexing daemon. Whenever an attribute in the database is defined as indexable for IR purposes (by appending a new tuple to *kw_sources*, a rule is created that appends the class name and attribute name to the *kw_index_flags* class whenever a new tuple is appended to the class. Another rule then starts the indexing process for the newly appended data. This trigger process is shown in Figure 3.

The indexing process extracts each unique keyword from the indexed attributes of the database, and stores it along with pointers to the record it came from along with the frequency of occurrence for that particular term in that field or document in *kw_term_doc_rel*. This process is shown in Figures 4 and 5. Other global frequency information is also maintained by the indexing daemon and the rules system, so that, for example, the overall frequency of occurrence of terms in the database, total number of indexed items, etc., are available for retrieval processing. The indexing daemon functions attempts to perform any outstanding indexing tasks before it shuts down. It also updates the *kw_doc_index* tuple for a given indexable class and attribute with a timestamp for the last item indexed. This permits ongoing incremental indexing without having to re-index older tuples.

4.2 Retrieval

The current version of the CSTR system uses a probabilistic retrieval algorithm that ranks retrieved documents in order of their estimated probability of relevance for a user's natural language query. The ranking algorithm that we are currently using is based on the *staged logistic regression* method described in (Cooper *et al.* 1992). From the user's point of view (or the interface programmer's point of view), the retrieval method is simply invoked as a function call (*kwsearch*)in a query language statement. The classes built by the indexing daemon are exploited to calculate the estimated probability of relevance for each indexed item in the database. Figures 6 and 7 shows the process involved in the probabilistic retrieval from the CSTR database.

The actual query to the Lassen retrieval process consists simply of a natural language statement of the searcher's interests. The query is treated much the same as the documents were treated in the indexing process. The individual words are extracted and matched to the *wn_index* dictionary (after removing stopwords). The unique termids for matching words from *wn_index* are then used to retrieve all the tuples in *kw_term_doc_rel* that contain the term. For each unique document ID in this list of tuples, the matching *kw_doc_index* tuple is retrieved. With the frequency information contained in *kw_term_doc_rel* and *kw_doc_index*, the estimated probability of relevance is calculated for each document containing at least one term in common with the query. The formulae used in the calculation and the sequence of operations performed to calculate the probability of relevance are shown in Figure 7.

Once the probability of relevance is calculated for each document, it is stored along with a unique query ID, the document ID, location information, in the *kw_retrieval* class. The query itself, along

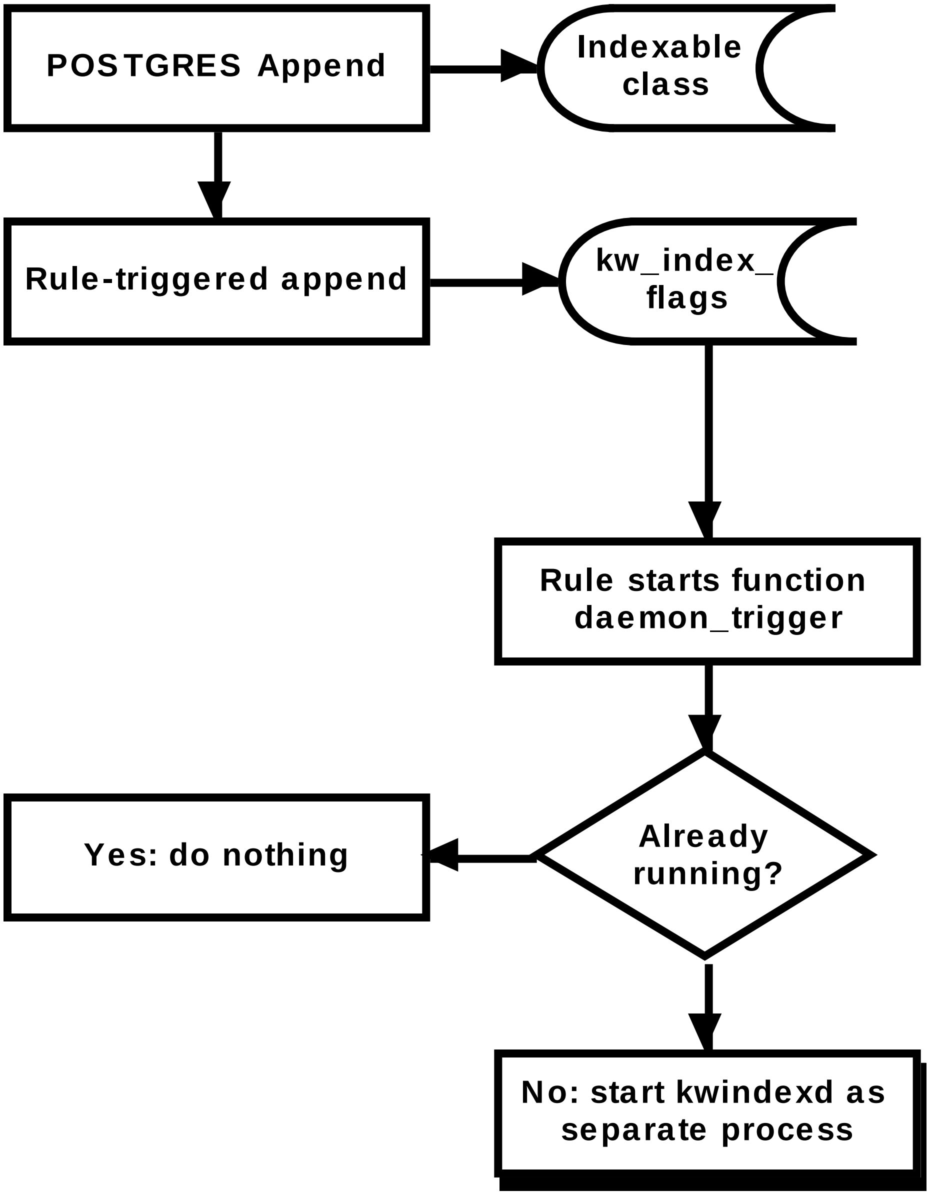

Figure 3: The Lassen Indexing Trigger Process

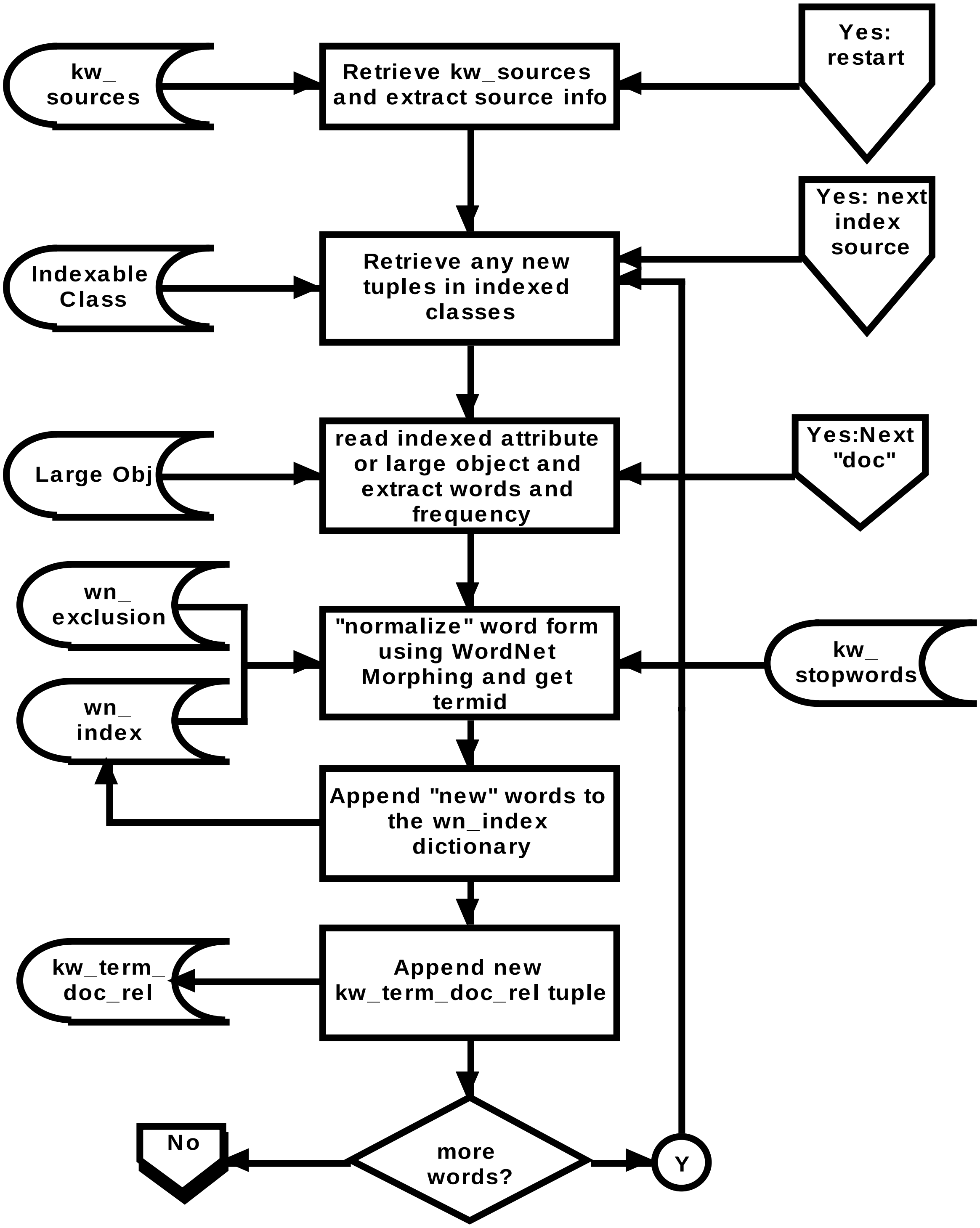
kw_ sources
Retrieve kw_sources and extract source info
Yes: restart
Indexable Class
Retrieve any new tuples in indexed classes
Yes: next index source
Large Obj
read indexed attribute or large object and extract words and frequency
Yes:Next "doc"
wn_ exclusion
wn_ index
"normalize" word form using WordNet Morphing and get termid
kw_ stopwords
Append "new" words to the wn_index dictionary
kw_term_ doc_rel
Append new kw_term_doc_rel tuple
No
more words?
Y

Figure 4: The Lassen Indexing Daemon Process

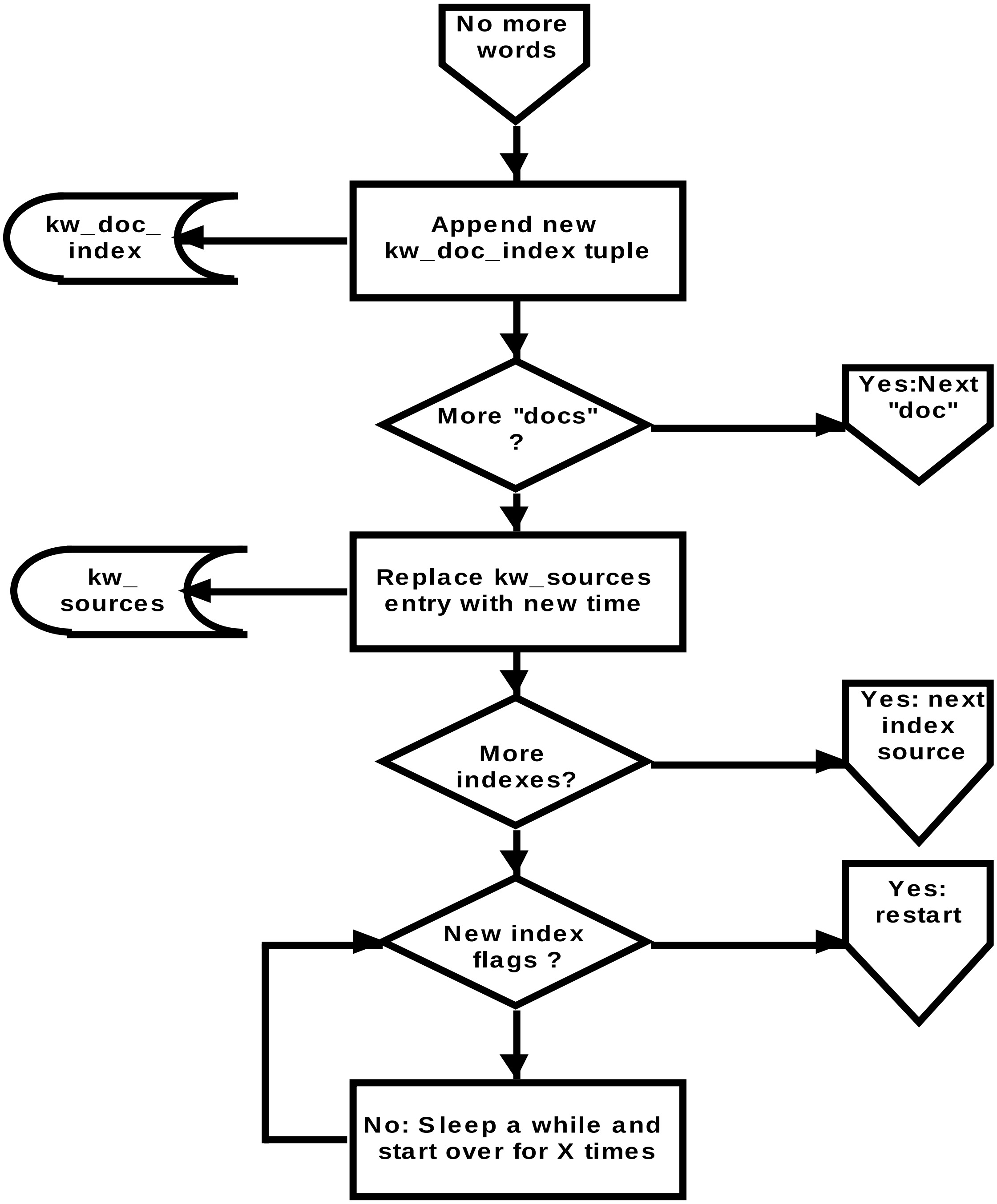

Figure 5: The Lassen Indexing Daemon Process (cont.)

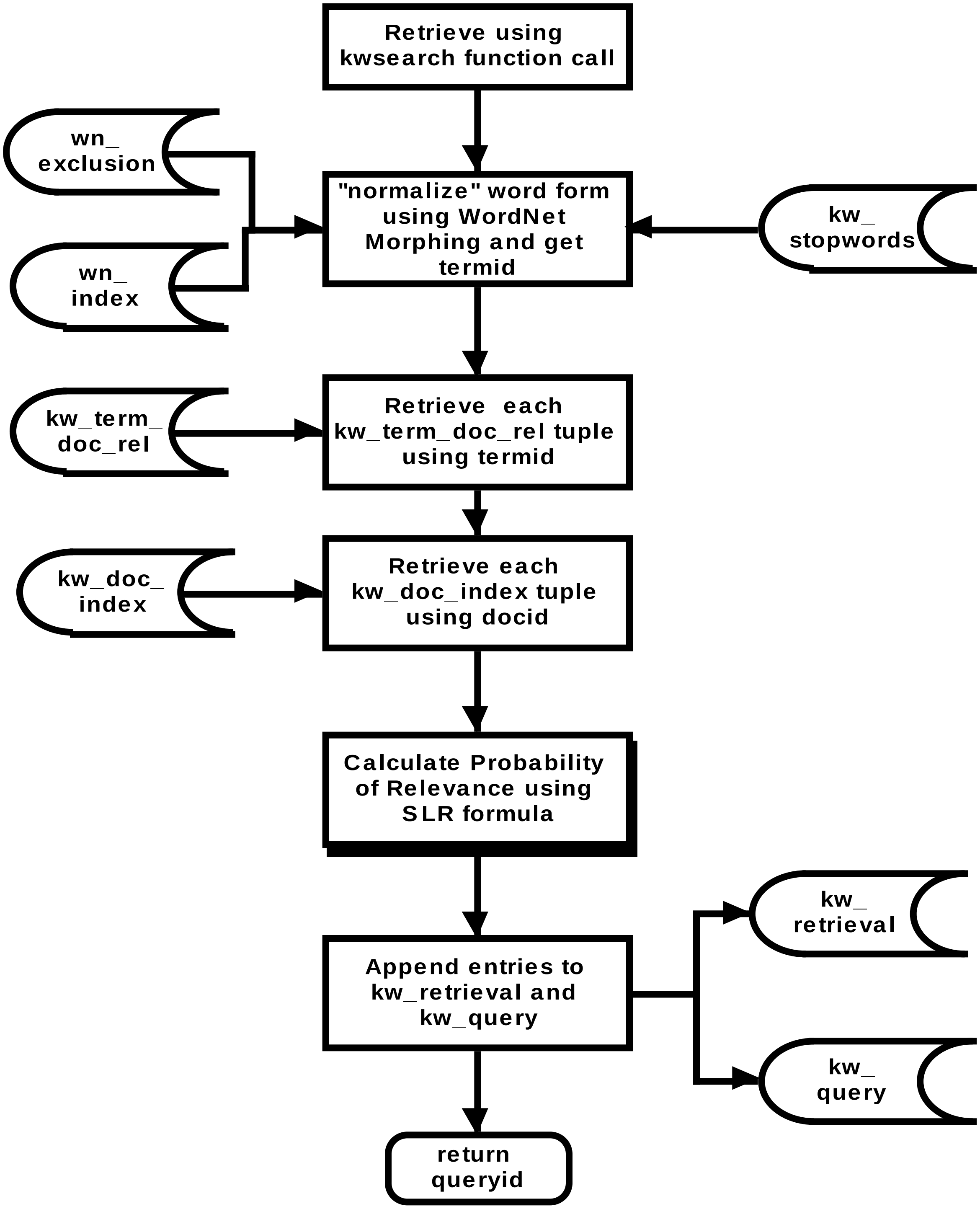
Retrieve using kwsearch function call
wn_ exclusion
wn_ index
"normalize" word form using WordNet Morphing and get termid
kw_ stopwords
kw_term_ doc_rel
Retrieve each kw_term_doc_rel tuple using termid
kw_doc_ index
Retrieve each kw_doc_index tuple using docid
Calculate Probability of Relevance using SLR formula
kw_ retrieval
Append entries to kw_retrieval and kw_query
kw_ query
return queryid

Figure 6: The Lassen Retrieval Process

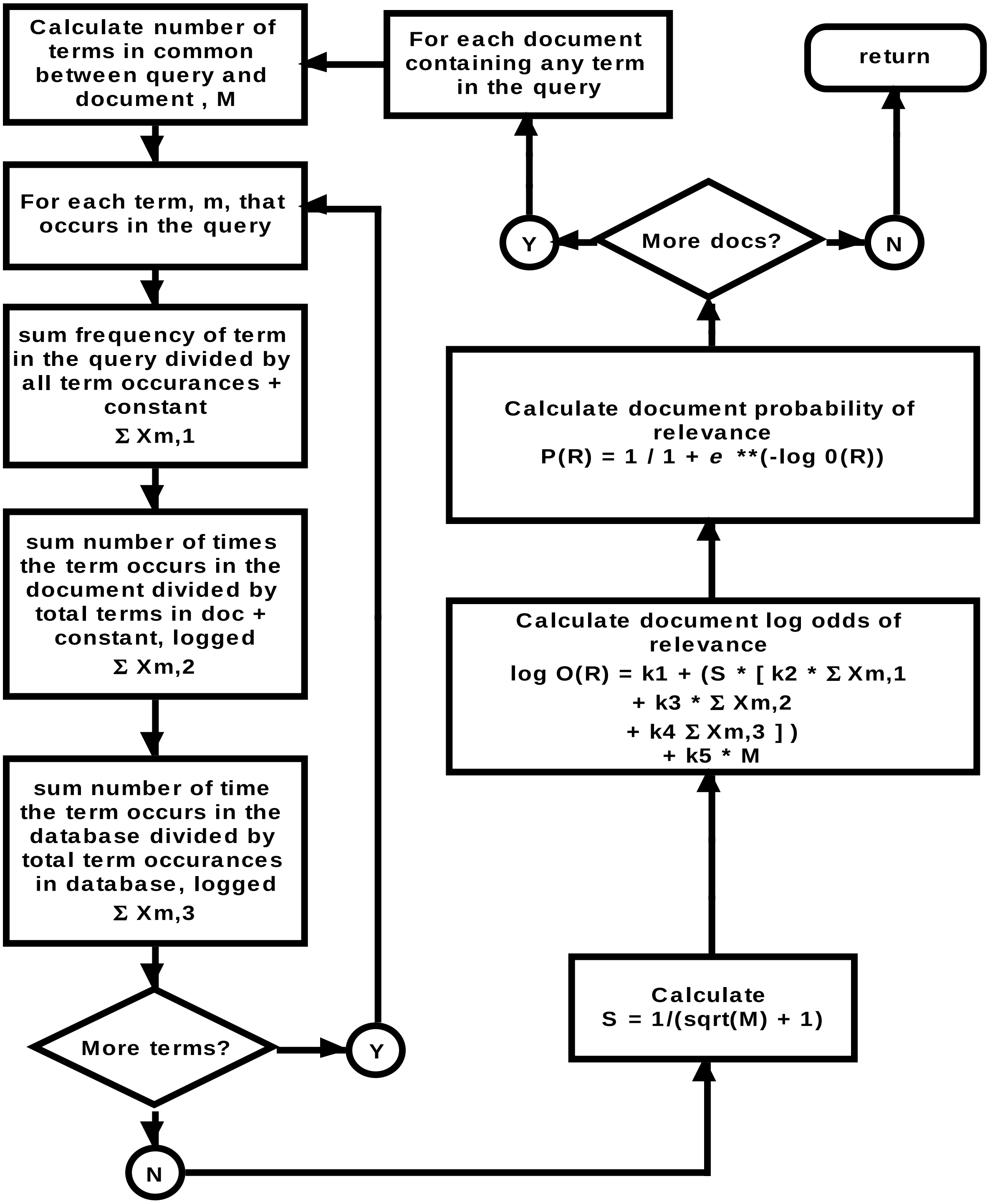
Calculate number of terms in common between query and document , M
For each document containing any term in the query
return
For each term, m, that occurs in the query
Y
More docs?
N
sum frequency of term in the query divided by all term occurances + constant
Σ Xm,1
Calculate document probability of relevance
P(R) = 1 / 1 + *e* **(-log 0(R))
sum number of times the term occurs in the document divided by total terms in doc + constant, logged
Σ Xm,2
Calculate document log odds of relevance
log O(R) = k1 + (S * [k2 * Σ Xm,1
+ k3 * Σ Xm,2
+ k4 Σ Xm,3])
+ k5 * M
sum number of time the term occurs in the database divided by total term occurances in database, logged
Σ Xm,3
Calculate
S = 1/(sqrt(M) + 1)
More terms?
Y
N

Figure 7: The Calculation for the SLR Probabilistic Ranking Process

with its unique id are stored in the *kw_query* class. To see the results of the retrieval operation, the query ID is used to retrieve the appropriate *kw_retrieval* tuples, ranked in order according to the estimated probability of relevance.

In the current design for indexing and retrieval operations, all of the information used is visible in user-accessible classes in the database. There is a fairly high overhead, in terms of storage and in processing time for maintaining the indexing and retrieval information in this way. We are investigating a class of new *access methods* to support indexing and retrieval in a more efficient fashion. This will involve declaring some POSTGRES functions that can extract sub-elements of a given type of attribute (such as words in a text string) and generate indexes for each of the sub-elements so extracted. Other types of data might also benefit from this class of access methods, for example functions that extract sub-elements like geometric shapes from images might be used to generate sub-element indexes of image collections.

5 Research Issues in Electronic Libraries

Beyond the ongoing development of a working electronic library, the Berkeley group is focussing on a number of research issues that will influence the future development of our own, and others, implementation of electronic libraries. The following briefly touches on the research agenda for the CSTR electronic library project at Berkeley. Most of these research efforts are currently underway, and haven't yet seen publication of results. Our primary research areas for the electronic library project are:

1. *To investigate exploiting natural language processing and artificial intelligence techniques to improve information access and retrieval.*

   The Berkeley CS department has a long history of work in natural language processing under Prof. Robert Wilensky, he is turning this expertise towards the problems of information retrieval. The Berkeley group is constructing a prototype Automated Librarian that incorporates uses natural language processing techniques to improve access to information in the repository. As part of this, work is underway on automatic classification of documents into categories using a combination of statistical and NLP methods.

2. *Design, implement and evaluate the viability of various types of user interfaces and retrieval paradigms for the electronic library.*

   As already seen, we are supporting a variety of user interfaces and retrieval methods for the electronic library project. We plan to evaluate these using methods ranging from transaction log analysis to user interviews and traditional experimental IR evaluation databases and methods.

3. *Integrate a DBMS view of electronic libraries with a communication networking view.*

   By supporting standards in both the DBMS and networking world we are attempting to provide a more comprehensive model for electronic libraries. We believe that many current network-based views for browsing and retrieval (such as WWW) will not scale well as the libraries grow in size and complexity. Database management systems were designed to scale to very large databases, but seldom possess the ease of use found in network-based systems. We are attempting to define an integrated view of electronic libraries that will scale, and be easy to use.

4. *Investigate techniques for organizing data in electronic libraries for presentation, including* virtual reality *browsing.*

   Researchers in the Berkeley group have developed an indexing tool that can segment a document into its significant subtopic sections(Hearst & Plaunt 1993). We plan to implement a new information access paradigm that uses this section information to provide enhanced information retrieval capabilities.

   We have also been developing a indexing method that can "read" the text of a document and automatically georeference any elements of the text that refer to places, providing the geographic coordinates of the point or polygon for the area discussed in the text(Woodruff & Plaunt 1994).

   In addition, we have been developing some new methods for automatically *categorizing* or identifying the major topics of a full-length document. This work (Hearst 1994) is still preliminary, but very promising results have been achieved for a large sample of the CSTR database. We hope to integrate the work on these indexing methods with the work on subelement access methods for the DBMS described above. This will provide a very powerful environment for experimentation in information retrieval, as well as support for the production CSTR electronic library.

   The Tioga interface is being developed to support a "joystick" interface that is intended to allow the user to pan and zoom through any conceivable combination of data elements as axes in a complex space.

5. *Investigate methods that will allow cost-effective search over a large number of widely distributed library sites.*

   As electronic libraries proliferate on the global network, the problem of distributed search becomes very important. How to decide which servers to search, whether to search in parallel or sequentially, and how to merge the results of distributed search are all research issues with no clear resolution. We are investigating the problems of distributed search, resource discovery, and how to merge result sets.

   We are developing several methods that address the problem of searching large numbers of distributed repositories. These methods include adaptive search algorithms, database-oriented architectural algorithms, and inductive learning algorithms. We will compare the methods and integrate the promising ones into a comprehensive distributed search strategy.

6 Conclusion

As electronic libraries proliferate, containing vast databases of information and linked together by the *information superhighway*, we believe that distributed, standards-based, scalable electronic libraries are inevitable, and that they must support all current and future media in an easily accessible and content-addressable fashion. To place this global virtual library at the fingertips of a world-wide clientel will require the development of intelligent client programs that can aid the user in exploring the thousands of distributed information servers. It will also require application of advanced techniques for information retrieval, information filtering, resource discovery, and the application of new techniques for automatically analyzing and characterizing data sources ranging from texts to videos. We see a need to develop technologies to:

- Provide a coherent, content-based view of a diverse distributed collection.

- Scale gracefully to very large (multi-terabyte) collections of databases.
- Facilitate data acquisition, transfer and presentation for information sources ranging from text to video

To this end, the Berkeley group has proposed a further research and development effort to continue the work begun on the CNRI CSTR project, under the NSF/NASA/DARPA Digital Libraries initiative. Electronic libraries are a fledgling technology with no firm standards, architectures, or even consensus notions of what they are and how they are to work. In the CSTR project at Berkeley, Stanford, CMU, Cornell, and MIT we are hoping to provide some of these standards, architectural models and definitions of the electronic library of the future.

References

Cohen, D., ed. (1992). A Format for E-mailing Bibliographic Records. Network Working Group Request For Comments: RFC1357, July 1992.

Cooper, W. S., Gey, F. C., & Dabney, D. P. (1992). Probabilistic Retrieval Based on Staged Logistic Regression. IN: *SIGIR '92 (Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Copenhagen, Denmark, June 21-24, 1992).* New York: ACM, p. 198-210.

Fox, E. A., et al. (1993). Users, User Interfaces, and Objects: Envision, a Digital Library. *Journal of the American Society for Information Science*, 44(8):480–491.

Hearst, M. A. & C. Plaunt (1993). Subtopic structuring for full-length document access. In *16th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, 27-30 June, 1993*, pages 59–68, New York. Association for Computing Machinery.

Hearst, M. A. (1994). Contextualizing Retrieval of Full-Length Documents. submitted to the *17th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*

Ousterhout, J.K. (1990). Tcl: an embeddable command language. IN: *Proceedings of the Winter 1990 USENIX Conference. (Proceedings of the Winter 1990 USENIX Conference, Washington, DC, USA, 22-26 Jan. 1990).* Berkeley, CA, USA: USENIX, p. 133-46.

Ousterhout, J.K. (1991). An X11 toolkit based on the Tcl language. IN: *USENIX Association. Proceedings of the Winter 1991 USENIX Conference. (USENIX Association. Proceedings of the Winter 1991 USENIX Conference, Dallas, TX, USA, 21-25 Jan. 1991).* Berkeley, CA, USA: USENIX Assoc, p. 105-15.

Stonebraker, M. & Kemnitz, G. (1991). The POSTGRES Next-Generation Database Management System. *Communications of the ACM*, 34(10): 78-92.

Stonebraker, M., Chen, J., Nathan, N., Paxson C. & Wu, J. (1993). "Tioga: Providing Data Management Support for Scientific Visualization Applications," IN: *Proc. 19th International Conference on Very Large Data Bases*, August 1993, Dublin, Ireland, p. 25-38.

Wilensky, R., Stonebraker, M., Larson, R., Buckland, M. & Lynch, C. (1992). A Proposal to CNRI for Research on Electronic Libraries. *unpublished manuscript.*

Woodruff, A. & C. Plaunt (1994). GIPSY: Georeferenced Information Processing SYstem. To appear in the Journal of the American Society for Information Science 1994.