Cheshire II: Designing a Next-Generation Online Catalog

Ray R. Larson, Jerome McDonough, Paul O'Leary, and Lucy Kuntz
School of Information Management and Systems, University of California, Berkeley, Berkeley, CA 94720-4600.
E-mail: ray@sherlock.berkeley.edu

Ralph Moon
The Library, University of California, Berkeley, Berkeley, CA 94720-4600

The Cheshire II online catalog system was designed to provide a bridge between the realms of purely bibliographical information and the rapidly expanding full-text and multimedia collections available online. It is based on a number of national and international standards for data description, communication, and interface technology. The system uses a client-server architecture with X window client communication with an SGML-based probabilistic search engine using the Z39.50 information retrieval protocol.

1. Introduction

Online public access catalogs have provided access to the collections of increasing numbers of libraries for over a decade. Indeed, many of the online catalogs at large research libraries are now over a decade old, and they have used the same basic search methods, user interface, and hardware configuration for that entire period.

While there have been various condemnations of online catalogs, or nostalgic recollections of card catalogs (Baker, 1994), the online catalog has been embraced by both librarians and library patrons (more or less happily). The primary effect of library automation, as applied to the catalog, has been to facilitate rapid and effective access to the desired items in the collection when the author, title, or subject headings of the item are known to the searcher. However, it has also been recognized for over a decade that the present generation of online catalogs in most libraries do not do a very good job of providing topical or subject access to the collections (Matthews, Lawrence, & Ferguson, 1983).

The common result of many subject searches (up to half of such searches in some systems) is search failure or "zero results." The reasons for this vary from search to search, but they include common misspelling, lack of knowledge of Boolean logic, and lack of familiarity with Library of Congress Subject Headings. When the user does succeed in a subject search, he or she will often be presented with far too many items to conveniently scan, in an order that has no relationship to the topical nature of works displayed or the query. The causes of this information overload vary from system to system and from search to search, but they include database size (or the relative collection size for a given topic) and the increasing numbers of items indexed by a given word in Boolean keyword-based online catalogs. The overload problem is compounded by the provision of keyword and heading truncation features in searching, and by searchers' tendency to use very general wording in their query formulations. At the same time, despite the problems they encounter in subject searching, online catalog searchers use subjects more frequently than any other access point in the catalog database. Many studies of online catalog use and users have found that regardless of the problems involved, subject or topical access is greatly desired and valued by online catalog users. (For a more comprehensive review of the research on subject searching, search failure, and information overload, see Larson, 1991a.)

The online catalog systems in place today are primarily derived from early information retrieval systems based on Boolean logic and exact keyword matching. Researchers have suggested that these systems are deficient for effective subject access for a variety of reasons (Hildreth, 1989). These include:

(1) They provide no aid to the searcher in formulating effective queries. There is usually no attempt to map from the searcher's notion of a topic to the terms or subject headings actually used to describe that topic in the database. Likewise, there are usually no facilities for broadening or narrowing the focus of a search.
(2) They do not foster browsing of the database. Most online catalogs have no way to exploit the many obvious linkages between database records (e.g., other

items with the same author, class number, or subject heading) without re-typing a complete query. Nor do they provide any simple facility for requesting "more like this one."

(3) They do not provide any useful ordering of retrieved records. Most online catalogs display retrieved records sorted into author/title order, regardless of the type of search performed. As we suggest below, a better ordering for topical searches is one based on the probability of relevance.

(4) They do not provide integrated access to information sources. Most online catalogs are strictly catalogs, and offer access to bibliographic records exclusively. While *digital libraries* (Fox, Akseyn, Furuta, & Leggett, 1995) have begun providing a wide variety of network-based information sources including full-text and multimedia information, the online catalog has not kept pace with the changing technology.

The Cheshire II project is focused on the development of a next-generation online catalog system that addresses these problems with existing online catalogs. In this article, we will describe the design of the Cheshire II system and its components. We also provide a more detailed discussion of some active research areas in the design and development of the search engine and methods for combining Boolean and probabilistic retrieval techniques.

2. The Cheshire II System

The Cheshire II online catalog system was designed to provide a bridge between existing online catalog technology and databases and the explosively growing realm of network-based digital libraries with information resources including full-text and multimedia. The primary objectives of the Cheshire II project have been: 1) To develop and demonstrate a next-generation online catalog system with advanced searching features using modern workstation and networking technology in a working library environment; and 2) to evaluate the retrieval performance and the use and acceptance of this online catalog by library patrons and remote network users.

Our design has been driven by the belief that many of the failures of subject access in online catalogs can be alleviated by removing some of the limitations on the computing technology, information retrieval methods, and user interaction techniques used in earlier online catalog systems. As pointed out above, most existing online catalogs are based on previous generations of computing technology, both hardware and retrieval algorithms, and do not take advantage of recent advances in computer hardware, software, and networking technology.

In the Cheshire II system, we have incorporated as many of these advances as possible in order to evaluate their effectiveness in the context of system performance, and in the use and usability of the system. The Cheshire project is intended to help understand and evaluate a number of practical and research problems in the design and evaluation of information systems. It provides both a practical demonstration of the use and effectiveness of an advanced online catalog system with "state-of-the-art" subject searching capabilities, and the utility of current standards for information retrieval and data structuring.

One goal of the system design was to provide an extensible system that can easily adapt to new types of data, and to provide a flexible and programmable user interface to display that data. In order to achieve this goal, we have attempted to incorporate appropriate national and international standards into the system wherever possible. The Cheshire II system design elements include:

(1) *An SGML Database:* SGML (Goldfarb, 1990) is used as the primary data base format of our underlying *search engine.*

(2) *Z 39.50 Client/Server Operation:* The system is based on a client/server architecture where the interfaces (clients) communicate with the search engine (server) using the Z39.50 Information Retrieval Protocol (ANSI/NISO, 1995).

(3) *Boolean and Probabilistic Searching:* The Cheshire II server (or search engine) supports both conventional Boolean and probabilistic "best match" ranked searching based on estimation of the probability of relevance for each query/document pair. The server permits Boolean and probabilistic elements within the same query.

(4) *An X Window-Based Graphical User Interface (GUI):* The user interfaces (Z39.50 clients) developed for the Cheshire II system provide a direct manipulation interface on X terminals. Support for Mosaic/World Wide Web access is under development.

(5) *Hypertext linkages and browsing:* The Cheshire II search engine and graphical user interface facilitate browsing through automatically generated hypertext links, and through "nearest neighbor" searches and relevance feedback.

There have been a number of experimental online catalog systems that have provided ranked retrieval (Fox, France, Sahle, Daoud, & Cline, 1993; Larson, 1992; Porter & Galpin, 1988; Walker, 1987; Walker, 1989), and a number of systems provide Z39.50 access, but the Cheshire II system is the first system to combine all of these design elements. In the following sections, we will discuss each of these design elements and the active research issues associated with them.

2.1. Cheshire II and SGML

In designing the Cheshire II system, we faced the question of how to provide a search engine that could be used on both simple text and complex structured records,

<USMARC Material="BK" ID="00000007"><leader><LRL>00893</LRL><RecStat>n</RecStat>
<RecType>a</RecType><BibLevel>m</BibLevel><UCP></UCP><IndCount>2</IndCount><SFCo
unt>2</SFCount><BaseAddr>00289</BaseAddr><EncLevel> </EncLevel><DscCatFm>a</DscC
atFm><LinkRec> </LinkRec><EntryMap><FLength>4</Flength><SCharPos>5</SCharPos><ID
Length>0</IDLength><EMUCP></EMUCP></EntryMap></Leader><Directry>0010014000000050
0170001400800410003101000140007202000150008603500200010103500170012103900180013 8
0400023001560500023001790820017002021000027002192450063002462600040003093000027 0
0349490003800376504005100414650002500465830003800490950003400528950003400562998 0
00700596</Directry><VarFlds><VarCFlds><Fld001>CUBGGLAD1258B</Fld001><Fld005>1994
0818092701.0</Fld005><Fld008>840216s1982 nyu b 00110 eng </Fld008></
VarCFlds><VarDFlds><NumbCode><Fld010 I1="Blank" I2="Blnk"><a>82016816 </Fld0
10><Fld020 I1="Blank" I2="Blnk"><a>0387907637</Fld020><Fld035 I1="Blank" I2=
"Blnk"><a>(CU)ocm08762561</Fld035><Fld035 I1="Blank" I2="Blnk"><a>(CU)GLAD12
58</Fld035><Fld039 I1="0" I2="Blnk"><a>23<c>3</c><d>3</d><e>3</e>
</Fld039><Fld040 I1="Blank" I2="Blnk"><a>DLC<c>DLC</c><d>OCL</d><d>CUY</d></
Fld040><Fld050 InLofC="Yes" CNSrc="Blnk"><a>QA274.2.E44 1982</Fld050>
<Fld082 Edition="Full" CNSrc="Blnk"><a>519.2/32<Two>19</Two></Fld082></NumbC
ode><MainEnty><Fld100 NameType="Single" I2=""><a>Elliott, Robert James.</Fld
100></MainEnty><Titles><Fld245 AddEnty="Yes" NFChars="0"><a>Stochastic calculus
and applications /<c>Robert J. Elliott.</c></Fld245></Titles><EdImprnt><Fld2
60 I1="" I2="Blnk"><a>New York, N.Y. :Springer,<c>c1982.</c></Fld260>
</EdImprnt><PhysDesc><Fld300 I1="Blank" I2="Blnk"><a>viii, 302 p. ;<c>25 cm.
</c></Fld300></PhysDesc><Series><Fld490 Traced="Differnt" I2="Blnk"><a>Applicati
ons of mathematics ;<v>18</v></Fld490></Series><Notes><Fld504 I1="Blank" I2=
"Blnk"><a>Includes bibliographical references and index.</Fld504></Notes><Su
bjAccs><Fld650 SubjLvl="NoInfo" SubjSys="LCSH"><a>Stochastic analysis.</Fld6
50></SubjAccs><AddEnty></AddEnty><LinkEnty></LinkEnty><SAddEnty><Fld830 I1="Blan
k" NFChars="0"><a>Applications of mathematics ;<v>18</v></Fld830></SAddEnty>
<HoldAltG></HoldAltG><Fld9XX><Fld950 I1="Blank" I2="Blnk"><l>ENGI</l><x>220</x><
a>QA274.2.E44 1982</Fld950><Fld950 I1="Blank" I2="Blnk"><l>MATH</l><x
>380</x><a>QA274.2.E44 1982</Fld950></Fld9XX></VarDFlds></VarFlds></U
SMARC>

FIG. 1. SGML version of USMARC record.

such as MARC and other bibliographic records, and also support complex multimedia documents and databases. After considering the variety of structured and unstructured data types that we intended to incorporate into the Cheshire II database, we decided to adopt the Standard Generalized Markup Language (SGML) (Goldfarb, 1990) as the fundamental data storage type for Cheshire II. All of the data in the Cheshire II database are stored as tagged, SGML documents (see Fig. 1 for an example of a MARC record in SGML format).

The adoption of SGML has provided a number of benefits for the Cheshire II system. The primary benefit has been that through the use of SGML tagging for all data in the database, and the adoption of the SGML Data Type Definition (DTD) language to define the structure of each data file, we have a common format for data types ranging from full-text documents, structured bibliographic records, to complex hypertext and multimedia documents (using the HTML DTD that defines the elements of World Wide Web (WWW) "pages"). This has important economies in the development process and in the addition of new types of data to the system.

Virtually all data manipulation for the database has been generalized as processes acting on SGML tags or sets of tags. Instead of having to develop new routines to manipulate each sub-element of a new data type, the developer only needs to provide a DTD and a conversion routine to convert the new data type to SGML. The built-in file manipulation and indexing routines can then extract and index any tagged sub-elements of the data-type for access. For example, after a MARC record such as the one shown in Figure 1 has been tagged, creating an index on a new element (such as keywords extracted from sub-tag "<a>" within tag "<FLD830>") simply involves specifying that within a configuration file, and running the extraction and indexing processes. Similarly, data extraction and indexing can be performed for any other tags specified in any SGML DTD, such as extracting the footnotes in a full-text document, or the captions of pictures in a WWW page.

SGML is also used as the basic format of Cheshire II configuration files. These files define the physical database elements of the Cheshire II system, including the locations of data files, which SGML DTD describes the file, and information on which indexes to create and the elements they should contain. Figure 2 shows a portion of a configuration file defining a MARC database, and

```
<!-- This is a sample configuration file for Cheshire II -->
<DBCONFIG>
<!-- The first filedef -->
<FILEDEF TYPE=SGML>
<!-- filetag is the "shorthand" name of the file -->
<FILETAG> bibfile </FILETAG>
<!-- filename is the full path name of the file -->
<FILENAME> /usr3/cheshire2/indexing/TESTDATA/morerecs.sgml</FILENAME>
<!-- fileDTD is the full path name of the file's DTD -->
<FILEDTD> /usr3/cheshire2/new/sgml/USMARC07.DTD </FILEDTD>
<!-- assocfil is the full path name of the file's Associator -->
<ASSOCFIL> /usr3/cheshire2/indexing/TESTDATA/morerecs.sgml.asso</ASSOCFIL>
<!-- history is the full path name of the file's history file -->
<HISTORY> /usr3/cheshire2/indexing/TESTDATA/morerecs.sgml.history</HISTORY>
<!-- The following are the index definitions for the file -->
<INDEXES>
...
<!-- Subject index definition -->
<INDEXDEF ACCESS=BTREE EXTRACT=KEYWORD NORMAL=STEM>
<INDXNAME> /usr3/cheshire2/indexing/TESTDATA/dictionary.subject </INDXNAME>
<INDXTAG> subject </INDXTAG>
<!-- The following INDXMAP items provide a mapping from the SUBJECT tag to -->
<!-- the appropriate Z39.50 BIB1 attribute numbers -->
<INDXMAP><USE> 21 </USE><POSIT> 3 </posit> <struct> 6 </struct> </INDXMAP>
<INDXMAP><USE> 26 </USE><POSIT> 3 </posit> <struct> 6 </struct> </INDXMAP>
<INDXMAP><USE> 25 </USE><POSIT> 3 </posit> <struct> 6 </struct> </INDXMAP>
<INDXMAP><USE> 27 </USE><POSIT> 3 </posit> <struct> 6 </struct> </INDXMAP>
<INDXMAP><USE> 28 </USE><POSIT> 3 </posit> <struct> 6 </struct> </INDXMAP>

<!-- The associator file for the index linking the termid with postings -->
<INDASSOC> /usr3/cheshire2/indexing/TESTDATA/mainfile.subj.idxasso </INDASSOC>
<!-- The postings file for the index containing all term/document/freq info -->
<INDXPOST> /usr3/cheshire2/indexing/TESTDATA/mainfile.subj.idxpost </INDXPOST>
<!-- The stoplist for this file -->
<STOPLIST> /usr3/cheshire2/indexing/TESTDATA/titlestoplist </STOPLIST>
<!-- The INDXKEY area contains the specifications of tags in the doc -->
<!-- that are to be extracted and indexed for this index -->
<INDXKEY>
<TAGSPEC>
<!-- Here we used wildcards '.'  to indicate all tags starting "FLD6" -->
<!-- followed by any two characters in the main file should be extracted -->
<!-- for this index -->
<FTAG>FLD6..  </FTAG>
</TAGSPEC> </INDXKEY> </INDEXDEF>
...
```

FIG. 2. Subject index from configuration file.

the index definitions for subject indexing in that file. Each database may contain multiple named files, and each of those may have any number of indexes extracted from them. The Cheshire II search engine relies on the configuration file to define all of the accessible elements of the database. Adding or deleting elements is as simple as changing the configuration file. Thus, it would be easy to change the Cheshire II system from an advanced catalog accessing a MARC database to a personal information retrieval system for a researcher's field notes, by making some changes to the basic configuration file.

2.2. The Cheshire II Search Engine

The original Cheshire catalog system was designed several years ago to test the use of probabilistic information retrieval methods upon MARC bibliographic data. In studies that compared these probabilistic retrieval algorithms to Boolean and other IR methods such as the vector space model, it was found that a combination of *classification clustering* and the probabilistic algorithms provided the best retrieval performance for a test database of MARC data (Larson, 1991b; Larson, 1992). In classification clustering, all of the title and subject words

for each record in a given class number are used to provide access points to that topical area.

For the Cheshire II project, the search engine was redesigned to support a variety of search and browsing capabilities. We have included facilities for both probabilistic and Boolean searching in Cheshire II. This was driven by the realization that there are different types of search tasks that are best handled by different retrieval methods. Therefore, we provide support for such methods as authority-controlled name searching and other conventional online catalog search features, such as "exact title" and "exact subject" matching capability and the ability to store and retrieve both Boolean and probabilistic "result sets" and use them in subsequent queries.

The search engine also supports various methods for translating a searcher's query into the terms used in indexing the database. These methods include elimination of unused words using field-specific stopword lists, particular field-specific query-to-key conversion or "normalization" functions, algorithms for reducing significant words to their *roots* or *stems* by converting suffix variations, such as plural forms of a word, to a single form, as well as support for mapping database and query text words to single forms based on the WordNet dictionary and thesaurus.

However, the primary functionality that distinguishes the Cheshire II search engine from conventional Boolean online catalog systems is the support for probabilistic searching on any indexed element of the database. This means that a natural language query can be used to retrieve the *best* matching records (or clusters, for clustered access points) in the database, and not just the exact Boolean matches. In both cluster searching and direct probabilistic searching of the database, the Cheshire II search engine supports *relevance feedback* so that any items found in an initial search (Boolean or probabilistic) can be selected and used as queries in a relevance feedback search.

This is an extension of the two-stage search method developed in the Cheshire prototype. In the prototype, probabilistic retrieval methods were used to match the searcher's query with a set of *classification clusters,* the searcher then selected the clusters that appeared relevant and they were combined with the initial query and used to re-rank the database, so that records were retrieved in decreasing order of probable relevance to the searcher's initial query statement, combined with the broad classes selected in the first stage. This two-stage search method appeared to assist the searcher in subject focusing and topic/treatment discrimination (Larson, 1991b). The cluster search method is still available in Cheshire II, but is now augmented by direct probabilistic searching of the database.

2.2.1. Probabilistic retrieval in Cheshire II. The probabilistic retrieval algorithm used in the Cheshire II search engine is based on the *staged logistical regression* algorithms developed by Berkeley researchers and shown to provide excellent full-text retrieval performance in the TREC evaluation of full-text IR systems (Cooper, Chen, & Gey, 1994a; Cooper, Gey, & Chen, 1994b; Cooper, Gey, & Dabney, 1992). Formally, the probability of relevance given a particular query and a particular document (i.e., record in the database) $P(R|Q, D)$ is calculated and the documents are presented to the user ranked in order of decreasing values of that probability. In the Cheshire II system $P(R|Q, D)$ is calculated as the "log odds" of relevance $\log O(R|Q, D)$, where for any events A and B, the odds $O(A|B)$ is a simple transformation of the probabilities $P(A|B)/P(\bar{A}|B)$. The Staged Logistic Regression (SLR) method provides estimates for a set of coefficients, c_i, associated with a set of S statistics, X_i, derived from the query and database, such that

$$\log O(R|Q, D) \approx c_0 \sum_{i=1}^{S} c_i X_i \qquad (1)$$

where c_0 is the intercept term of the regression.

For the set of M *terms* (i.e., words, stems, or phrases) that occur in both a particular query and a given document, the equation used in estimating the probability of relevance for the Cheshire II search engine is essentially the same as that used in (Cooper et al., 1994b) where the coefficients were estimated using relevance judgements from the TIPSTER test collection:

- $X_1 = (1/M) \sum_{j=1}^{M} \log QAF_{t_j}$. This is the log of the absolute frequency of occurrence for term t_j in the query averaged over the M terms in common between the query and the document. The coefficient c_1 used in the current version of the Cheshire II system is 1.269.
- $X_2 = \sqrt{QL}$. This is square root of the query length (i.e., the number of terms in the query disregarding stopwords). The c_2 coefficient used is −0.310.
- $X_3 = (1/M) \sum_{j=1}^{M} \log \mathrm{DAF}_{t_j}$. This is is the log of the absolute frequency of occurrence for term t_j in the document averaged over the M common terms. The c_3 coefficient used is 0.679.
- $X_4 = \sqrt{DL}$. This is square root of the document length. In Cheshire II, the raw size of the document in bytes is used for the document length. The c_4 coefficient used is −0.0674.
- $X_5 = (1/M) \sum_{j=1}^{M} \log \mathrm{IDF}_{t_j}$. This is is the log of the *inverse document frequency* (IDF) for term t_j in the document averaged over the M common terms. IDF is calculated as the total number of documents in the database, divided by the number of documents that contain term t_j. The c_5 coefficient used is 0.223.
- $X_6 = \log M$. This is the log of the number of common terms. The c_6 coefficient used in Cheshire II is 2.01.

The Cheshire II search engine calculates all matching functions at the point of retrieval, rather than pre-computing portions of the functions. Only the fundamental statistics (such as raw term frequency) are stored in the

database, making it easy to apply a different algorithm to the same database without re-indexing.

Probabilistic searching, as noted above, requires only a natural language statement of the searcher's topic, and thus no formal query language or Boolean logic is needed for such searches. However, the Cheshire II search engine also supports complete Boolean operations on indexed elements in the database. One active area of research is examining the combination of Boolean and probabilistic ranked elements within the same query, which we discuss in the following section.

2.2.2. Combining Boolean and probabilistic searching. The Cheshire II system provides users with the ability to search using either natural language queries with probabilistic ranking of search results or conventional Boolean queries and term matching, as well as the option to use both types of searches simultaneously. Although these are implemented within a single process, they comprise two parallel *logical* search engines. Each logical search engine produces a set of retrieved documents. When a user chooses only one type of search strategy, then the result set of that search is presented directly to the user, either a probabilistically ranked set or an unranked Boolean result set. When the user queries the database using the parallel search strategies, the two result sets are merged and presented to the user as a single set.

Each of these two types of queries can be thought of as distinct representations of the user's abstract information need—each with advantages for particular types of searches. The parallel querying process allows the user to state the information need in more than one form, thus giving the whole system a more complete statement of that need. This also allows users to take advantage of the strengths of each search strategy and to create queries tailored to their particular requirements. For example, in searching for a known item or known author, explicit Boolean query formulations are effective. Alternately, in subject searching, when users rarely know the indexing or classification terms used to describe the desired but unknown items, probabilistic matching of queries to documents is more effective.

From the user's perspective, however, there may not be a clear distinction between these types of searches. Users of ranked retrieval systems (usually those experienced with Boolean systems) have often expressed a desire to refine a ranked retrieval by using the more restrictive Boolean operators in conjunction with the ranking mechanism. This combination of retrieval methods would allow the user, for example, to disambiguate the sense of a keyword in the Boolean query with a description of its intended sense or context in the probabilistic query. In effect, the system would raise the relevance ranking of documents associated with the desired sense of the exact match keyword. Another example of the usefulness of this parallel search strategy is the case where the documents retrieved under the heading of an especially prolific corporate author were ranked according to a probabilistically defined topic statement. In each case, the user is able to specify a particular information need more precisely and to retrieve a better ranking of relevant documents, than either one of the two types of queries and search strategies would afford.

Besides allowing the user greater flexibility, the motivation for using two search methods follows from the observation that no single retrieval algorithm has been consistently proven to be better than any other algorithm for all types of searches. By combining the retrieved sets from these two search strategies, we hope to leverage the strengths and reduce the limitations of each type of retrieval system. In general, the more evidence the system has about the relationship between a query and a document, the more accurate it will be in predicting the probability that the document will satisfy the user's need (Belkin, Kantor, Cool, & Quatrain, 1994). Other researchers (Keen, 1992) have shown that additional information about the location and proximity of Boolean search terms can be used to provide a ranking score for a set of documents. Recent IR models have shown that the exact match Boolean retrieval status can be used as evidence of the probability of relevance in the context of a larger network of probabilistic evidence (Fuhr, 1992; Turtle & Croft, 1990; Turtle & Croft, 1992). In the same way, we treat the set of documents resulting from the exact match Boolean query as a special case of a probabilistically ranked set, with each retrieved document having an equal rank. The Boolean result set is combined with the ranked result set from the probabilistic query to form a single ranked result set using evidence from both logical retrieval engines to determine a more accurate probability of relevance.

2.2.3. Merging the results of two logical search engines. The primary problem in this parallel retrieval strategy is in determining the relationship between the results of the two retrieval systems. This relationship can be seen as the dependency between two types of evidence in a probabilistic inference network (Fuhr, 1992; Turtle & Croft, 1990). The dependency cannot be formally specified before an actual query is submitted because the system cannot predict the features of the particular query. For example, if the user enters the same term in each side of the parallel query, then the result sets could be expected to have a very high degree of statistical dependence: The presence of a document in the exact match result set predicts a higher rank for that document in the probabilistic result set. Alternately, if the user submits terms from widely divergent subject areas on each side of a parallel query, then the result sets would be expected to have some lesser dependency relationship; the retrieval of a document by exact match to the one term predicts little or nothing about that document's ranked relevance to the other term.

Another theoretical issue involves the difference in the underlying models of exact match and probabilistic retrieval. In its basic form, the Boolean exact match model is only concerned with the simple matching of query terms to document or index terms; the model says nothing about the user's subjective relevance judgments. Probabilistic models, on the other hand, include a model of the user's judgments by attempting to predict relevance with statistical evidence. Complicating things further, there are two general interpretations of the ordering of probabilistic ranked sets (Bookstein, 1985; Fox & Koll, 1988). One treats the document rankings as representing degrees of relevance. The weight or rank assigned to any document in the retrieved set reflects a relative measure of the document's relevance. In the other interpretation, the rank of each document represents a probability that a document is relevant in absolute terms. If we focus on the user's perspective, however, this distinction is not critical because users of IR systems tend to interpret probability rankings as relative relevance rankings, or measures of potential usefulness in satisfying the need expressed in the query (Bookstein, 1983). Extending this idea from the user's point of view, we can consider membership in the Boolean retrieved set as a prediction of a user's relevance judgment. This is in keeping with Turtle and Croft's incorporation of the Boolean retrieval model into the probabilistic inference network model (Turtle & Croft, 1990; Turtle & Croft, 1992). We will merge the result sets of our Boolean and probabilistic search engines with the justification that Boolean retrieval is a special case in the probabilistic model.

In order to merge the two sets into one ranked set, we need to anticipate some relationship between the two sets. As discussed above, we cannot predict the dependency relationship between the terms in each part of the query. But we can specify a relationship between the Boolean and probabilistic results based on a simple analysis of the Boolean portion of the query. This analysis classifies the Boolean query, and by extension the Boolean retrieved set, based on the relative advantages and disadvantages of an exact match search strategy compared to a probabilistic search strategy. We then assign relative value to the Boolean retrieved set, in the form of a coefficient in the merging algorithm.

The objective is to weight the result sets based on the type of search, favoring each system or retrieval strategy where it is most effective. For known item searches, such as author or exact title searches, the Boolean retrieval set is given the highest weighting in the merging process. The Boolean retrieved sets from title or abstract keyword searches are assigned less of a value than those from known item searches. As a starting point for testing, results from these queries are given equal value with the results of probabilistic queries. Finally, result sets from Boolean keyword searches in the full text of a document—the lowest precision search in a Boolean system—are assigned weights of less value than the result set of a probabilistic query. In this case, the merged result set weighted in favor of the more useful probabilistic search engine, and augmented somewhat by the occurrence of a keyword in the text of the document. We expect this method of merging the results of Boolean and probabilistic queries to be especially useful in improving the results of this type of keyword Boolean retrieval strategy. These merging strategies, and the coefficients for merging different types of searches, are being evaluated for future implementation if testing shows a significant improvement over the current strategy.

At present, combined probabilistic and Boolean search results are evaluated using the assumption that the Boolean retrieved set has an estimated $P(R|Q_{\text{bool}}, D) = 1.0$ for each document in the set, and 0 for the rest of the collection. The final estimate for the probability of relevance used for ranking the results of a search combining Boolean and probabilistic strategies is simply:

$$P(R|Q, D) = P(R|Q_{\text{bool}}, D)P(R|Q_{\text{prob}}, D) \quad (2)$$

where $P(R|Q_{\text{prob}}, D)$ is the probability estimate from the probabilistic portion of the search, and $P(R|Q_{\text{bool}}, D)$ the estimate from the Boolean. This has the effect of restricting the results to those items that match the Boolean portion, with ordering based on the probabilistic portion.

2.2.4. Browsing and relevance feedback. One desirable property of an online catalog is to provide methods of open-ended, exploratory browsing through the database. This feature is being implemented in the Cheshire II search engine is several ways.

One obvious way to provide browsing is to permit the user to follow static or dynamically established linkages between records in the database, (e.g., jump to the next record with the same subject heading) in order to find items with some association to those previously retrieved. In Cheshire II, this functionality is being provided by a combination of selection mechanisms in the user interface and exact Boolean search methods in the search engine (this hypertext mechanism is described further in the following section).

A more interesting method for browsing is the inclusion of relevance feedback in the Cheshire II search engine. In the current implementation, relevance feedback is implemented as probabilistic retrieval based on extraction of content-bearing elements (such as titles, subject headings, etc.) from any items that have already been seen and selected by a user. Thus, any citation or document seen by the user can become the basis for a *nearest neighbor* search, where it is used as a query to find those records in the database most similar in content to the one specified. Similarly, multiple records may be selected and submitted for feedback searching. In this case, the contents of all those records are merged into a single query and submitted for searching. In the current imple-

mentation, generating a feedback search is accomplished by parsing the selected record(s) and extracting the record elements specified for the index used for topical searching (as specified in the database configuration file). Each of these record elements is combined to form a single query, which is then submitted to the same probabilistic retrieval process described above. At the present time, we do not use any methods for eliminating poor search terms from the selected records, nor special enhancements for terms common between multiple selected records (Salton & Buckley, 1990), but we plan to experiment further with various enhancements to our relevance feedback method.

2.3. The Cheshire II Client Interface

The evolution of the Cheshire II client interface has been driven by a tension between two desires on the part of the designers. The first of these desires was to produce a client interface that was more than simply a GUI for traditional OPAC searching; we hoped to produce a client which would support end-user searching with a variety of Z39.50 servers, any of which might support many different search engines and produce several different document formats. Our second desire was, to the extent possible, to minimize the cognitive load on users wishing to search this diverse set of resources by providing a single, coherent user interface for interacting with all of them. Our hope was to produce a client capable of searching either an OPAC system or an image database (and displaying results from those searches) with equal facility and with minimal reconfiguration of the interface itself.

There were several other design criteria that we formulated for the client interface. While we hoped to limit reconfiguration of the interface as the user moved from server to server, we also wanted to ensure that screen space was not wasted in presenting mechanisms for search interaction that were irrelevant in the context of a particular client-server session. As an example, if a user switched from a search session with the Cheshire II server to one with the University of California Melvyl Z39.50 system, those aspects of the interface necessary for specifying probabilistic queries would no longer be useful and should be removed from the display. Obviously, our hopes in this regard were to some degree in direct conflict with our desire to minimize changes to the interface when moving from server to server, and negotiating between these goals has proved one of the more difficult aspects of the client design. In addition to the goals already stated, we also hoped to:

(1) Minimize use of additional windows during users' interactions with the client in order to allow them to concentrate on formulating queries and evaluating the results, and not expend additional mental effort and time switching their focus of attention from the search interface to display clients;
(2) provide functions not immediately related to searching, such as print and E-mail facilities, to facilitate users' ability to "take the results home"; and
(3) design a help system within the interface that would assist users not only in the mechanics of operating the Cheshire II client, but also in the more general tasks of selecting appropriate resources for searching, formulating appropriate queries, and employing various search tactics.

In particular, by monitoring users' search results and, when possible, providing context-sensitive suggestions on how to improve a query, we planned to provide an interface that would assist users in both refining a search over time and extracting useful information as their search progressed, as suggested by Bates (Bates, 1989).

To date, we believe we have been reasonably successful in negotiating among these goals. Figures 3 and 4 show the reconfigurations to the interface which occur in switching from a search with the Cheshire II server (where a mixed probabilistic and Boolean search is being performed) to one with Melvyl Z39.50 server, which only supports Boolean queries. The text entry area and ranking type selection button for specifying probabilistic queries is removed, and in its place two additional Boolean index specification/text entry areas are provided. The mechanism for selecting a Boolean index (a pull-down menu to the left of the applicable text entry area) is the same in both instances, although the list of indexes is altered to reflect the indexes available with the current server. The mechanism for specifying Boolean operators (AND/OR buttons between the text entry areas) is also the same, although two additional buttons are provided in the pure Boolean interface to enable more complex Boolean queries.

This reconfiguration of index names and searching features will use the Z39.50 v.3 "Explain" database to discover the characteristics of the server and adapt the interface to it. The Explain database is a special database with record formats and searchable elements defined in the Z39.50 v.3 standard (ANSI/NISO, 1995; Lynch, 1995). It contains information about the server, its databases, and search elements available in those databases, in a standard machine-readable form. For older servers with no Explain database, manually constructed tables of information about the server are used in Cheshire II for known databases.

One additional interaction feature to be noted in the figures is the dynamically generated hypertext links associated with each name and subject heading in the displayed records (indicated by raised and highlighted text). Each of these is a button that submits a Boolean query consisting of the highlighted text with the appropriate index specification. This dynamic hypertext mechanism is based on the client's ability to identify these elements in both SGML records from the Cheshire

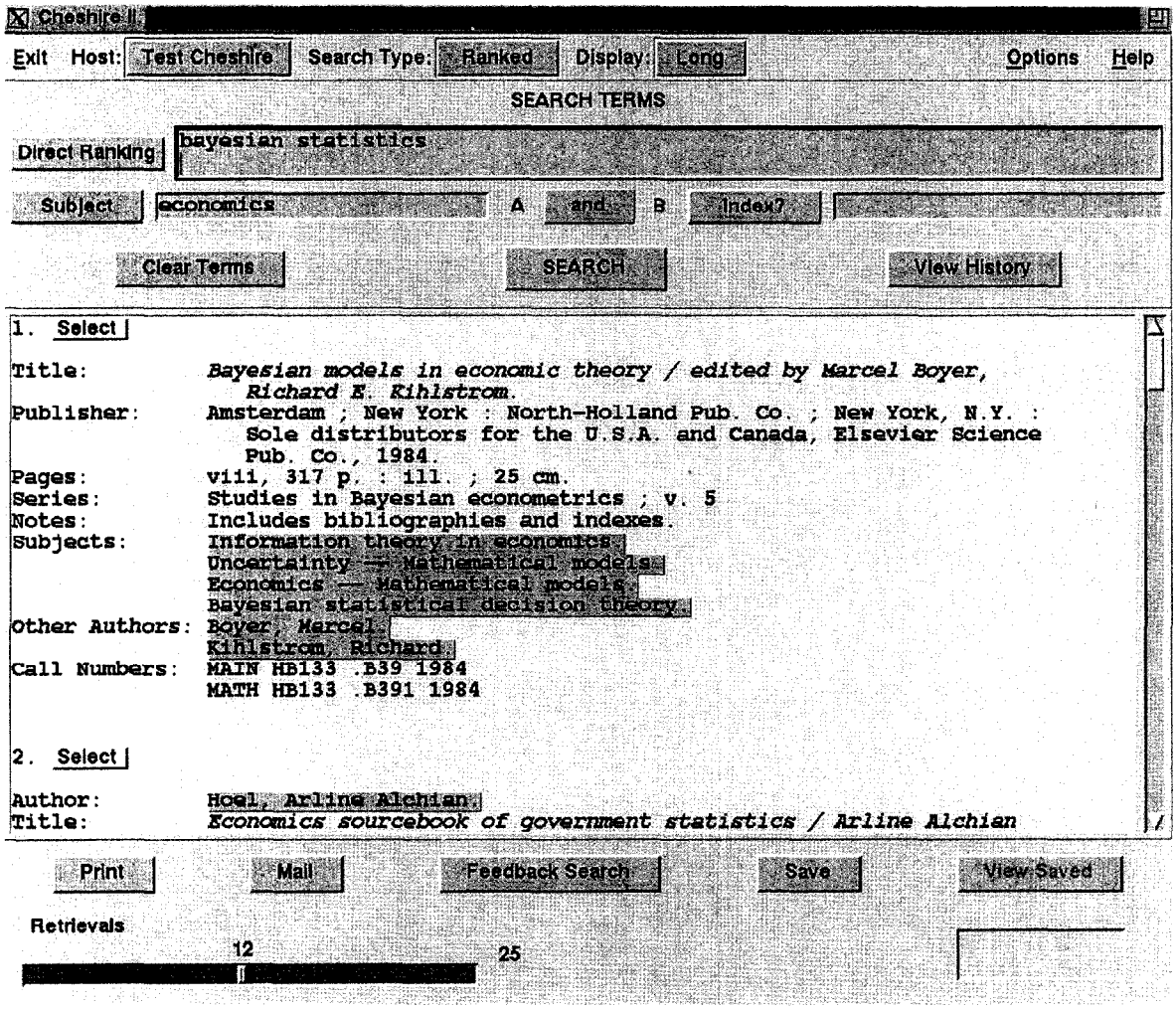

FIG. 3. Cheshire II client performing mixed probabilistic/Boolean search.

II server, and in MARC records from other Z39.50 servers. The client software can then treat each of the texts as a button and associate an action (submitting a new query) with each one. This permits very simple browsing of the database by following subject heading or author links.

Additional functionality beyond searching and browsing has been relatively easy to implement. Functions for printing, E-mailing, and saving records are all available when records are displayed, and the user has the option of acting on either the entirety of the current record display or a subset thereof by selecting individual records using the "select" buttons on each record (visible in Fig. 3 next to the record numbers). Figure 5 shows the client's E-mail facility, which includes the ability for the user to provide additional text in forwarding selected records to a particular E-mail account.

The Cheshire II client interface has been primarily implemented using the interpreted Tcl/Tk language (Ousterhout, 1994), with a variety of lower-level functions, including the majority of the Z39.50 client interactions, written in the C programming language. This combination has proven quite successful in both providing the ability to rapidly prototype and modify the graphic user interface to accommodate new features (such as the result summarization and reporting found in the OASIS system (Buckland, Butler, Norgard, & Plaunt, 1993), and maintain a relatively high level of performance for the Z39.50 client-server interactions. The combination of Tcl/Tk and the workstation hardware being used for the evaluation experiments permits the use of multimedia information sources including graphics and sound, and will permit display of mathematical formulae and non-roman characters. We are also considering altering the existing help facilities, which use Tcl/Tk text-tagging features for enhanced graphic display and hypertext links, to support display of SGML documents.

In addition to the Cheshire II client interface, complete access to the Cheshire II server is available through other Z39.50 clients. The Cheshire II server also provides support for the HTTP protocol via an HTTP-to-Z39.50 gateway, giving access to popular WWW clients like Mo-

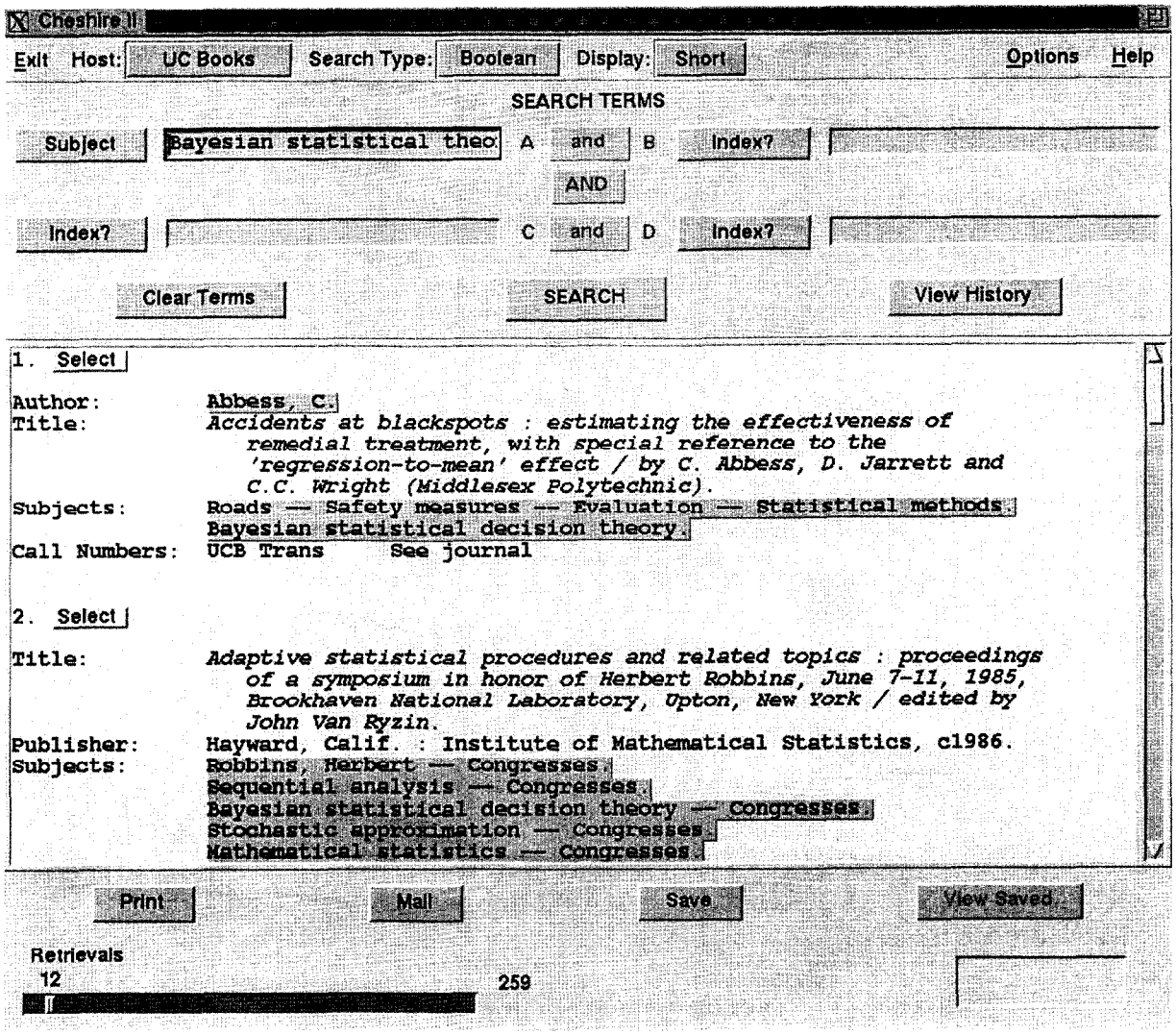

FIG. 4. Cheshire II client performing pure Boolean search.

saic and Netscape. This interface (using HTML forms for data entry elements) provides remote network users many of the same search features as the full client described above, with some loss of integration and ease of interactivity. Because HTTP is a stateless protocol, with each query/response pair considered a complete transaction, the ability to do relevance feedback is very limited in the current WWW implementation.

3. Evaluation Objectives

One of the primary goals of the Cheshire II project has been to produce a system that can be used in an actual library setting, and to evaluate the user's behavior with, and responses to the system (particularly with regard to its advanced retrieval methods). In evaluating a system like Cheshire II, there are several different aspects to consider. First, there is the performance of the system itself. This includes both efficiency and effectiveness. Next, there is the user interface and how well it functions. Finally, there is the user, and determining user satisfaction and search patterns.

Each category breaks down further into specific evaluation goals. The efficiency of the system will be measured in terms of its response time. That is, how long it takes between the time a query is entered and the time results are displayed. Evaluation of system effectiveness will be based on calculations of precision and recall using standard IR test collections and also by using selected queries from users and expert evaluation of search results. In addition, overall user satisfaction will be considered. Another potential measure of system effectiveness will be calculated using the proportion of records in a result set which are saved or sent to the user through electronic mail. This is a crude, but potentially helpful way to estimate the usefulness of the records to the user.

The issues surrounding evaluation of the interface include the ease with which users learn how to use the system and how well the users can accomplish their tasks.

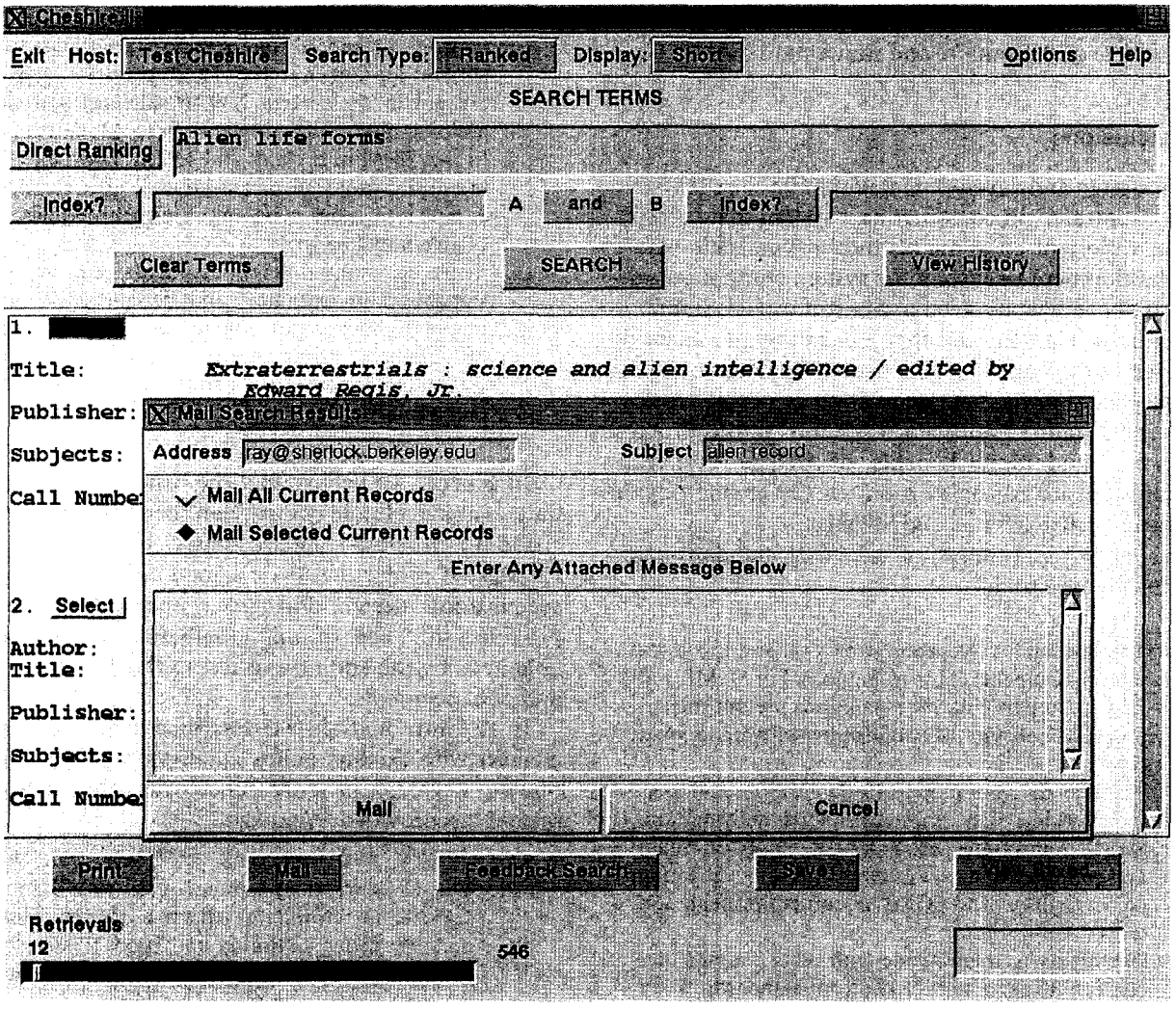

FIG. 5. Cheshire II client E-mail facility.

We will also evaluate the help system and how easily users can correct their mistakes. The system will also be available to network users via WWW browsers such as Mosaic or Netscape, which do not provide the primary graphical interface discussed above. This will allow us to evaluate the interface features presented by by means of comparing the experiences of both types of users.

The search patterns of the users are of great interest. Demographic information such as age, gender, and academic area will be collected in order to explore possible differences in searching styles, success, and satisfaction. In addition, the overall relative use of the different search capabilities will be determined. That is, the amount of Boolean searching will be compared to probabilistic searching and the use of searching clusters will be compared to direct ranking. Note that this information will come from direct observation, via the transaction logs, and will not depend on the user knowing what type of search is being done. The usefulness of the various indices is also of interest. Frequency of use, search results, and user satisfaction in this context will all be examined.

Two primary methods for evaluation will be used. The first involves transaction monitoring and logging of significant events in the users' interaction with the system. These transaction logs are recorded automatically by the system (at both the server or search engine, and in the client for local users). The second method is an online questionnaire presented for users to complete at the end of a search session. With questionnaire administration handled entirely online, network users at remote locations can participate in the evaluation of the system and its use. This would be much more difficult to accomplish with a paper questionnaire.

The main drawback to these evaluation methods is that there is no direct contact with the user. Thus, there is no way to gain insight into the thought processes of the user or any other background information not specifically requested in the questionnaire. We plan to remedy

this lack with interviews of a subset of local users to supplement the questionnaire and transaction data.

4. Conclusions

The design and development of the Cheshire II system has concentrated on constructing a system that incorporates a variety of components into a synergistic whole. The development of each of the system components described above has involved a taking a model, standard, or prototype element and then extending and adapting that element to confirm to an overall structure composing our vision of the next generation of online catalogs and similar online information systems. In this process, we have found many benefits, as well as numerous difficulties, from basing portions of the technology on standards such as Z39.50 and SGML.

The principle benefits of adopting standard-based technology have been:

(1) The availability of precise and exacting specifications for elements of the technology. For SGML and Z39.50, in particular, the standards present the appropriate behavior of conforming software in great detail.
(2) The availability of supporting applications and tools for working with standards-based information. SGML, for example, has a number of public domain and commercial tools available including validating parsers, editors and SGML document presentation tools.
(3) The ability to interoperate with other systems that conform to all, or part of, the same standards. As an example of this, we were able to begin development of our user interface while the search engine was still in development by using Z39.50 to interact with other search engines over the Internet.
(4) Standards-based technology can be more easily shared with others, and those working within a standardized framework benefit from a wider community of users and developers working on similar problems.

There are also some drawbacks to using standards-based technologies in the design and development process.

(1) Standards like SGML and Z39.50 are complex, and developing a system that conforms to these standards is a much more time consuming and difficult task than it would be to develop a non-conforming and non-standard system.
(2) Standards are evolving over time, and thus offer something of a "moving target" for developers. This often raises the issue of whether the system should conform to a previous version of a standard, or attempt to support the incompletely defined "next version."
(3) Not all desirable features of an information retrieval system like Cheshire II are supported in the current version of standards like Z39.50. In particular, full support for ranked retrieval and relevance feedback are not in the current Z39.50 standard and were added as non-standard extensions to Z39.50 support in the Cheshire II client and server. Thus, to be an interesting research system as well as a standards compliant system, while not antithetical goals, are often competing goals.
(4) There are many different standards (and de facto standards) in existence, and designers are often forced to choose among them. There is some danger of making a "wrong choice" and being left with a system that is completely compliant with a standard that nobody uses.

In general, we believe that the benefits of using standards-based technology outweigh the problems. The decision to develop standards-based technology in the Cheshire II project has been a good approach to system specification, design, and development. The standards provide a solid base of functionally that is extended and enhanced by the addition of research-driven extensions and enhancements.

In the introduction, we described a number of deficiencies with existing online catalog systems in most libraries. For each of those deficiencies, the Cheshire II system has provided a remedy:

(1) The Cheshire II system attempts to aid to the searcher in formulating effective queries. This is done by using stemming and probabilistic "best match" algorithms, and by the use of classification clusters to help focus topical searches.
(2) The system fosters browsing of the database. We provide relevance feedback and "nearest neighbor" searching for any record displayed to the user. We also provide "point and click" hypertext searching in the user interface, to retrieve items with the same authors or subjects as those selected.
(3) The system provides an ordering of retrieved records in topical searches based on the estimated probability of relevance.
(4) The system provides support for a wide variety of data types stored as tagged SGML records. This provides a general search and retrieval engine that can be used for complete digital library systems (Fox et al., 1995) including full-text and multimedia information, and not just the online catalog.

As of this writing, the Cheshire II project is actively researching many of the issues described in this article. We are combining work on database structures and algorithms for probabilistic information retrieval, advances and extensions to standard information retrieval protocols, graphical user interfaces, and user evaluation in a single project. The Cheshire II system is being installed in the UC Berkeley Mathematics, Statistics, and Astronomy library and will soon be available to library users.

We plan to publish further descriptions of our findings on retrieval algorithms, use of SGML structured documents as database objects, user interfaces, and user reactions to the Cheshire II advanced online catalog.

5. Acknowledgments

The work described in this article was sponsored by a College Library Technology and Cooperation Grants Program, HEA-IIA, Research and Demonstration Grant (R197D30040) from the U.S. Department of Education. The authors also thank the anonymous reviewers for helpful suggestions on improving this article.

References

ANSI/NISO Z39.50-1995. (1995). *Information retrieval (Z39.50): Application service definition and protocol specification (ANSI/NISO Z39.50-1995).* Bethesda, MD: NISO Press. Available: ftp://ftp.loc.gov/pub/z3950/official.

Baker, N. (1994). DisCards. *New Yorker, 70*(7), 64–86.

Bates, M. J. (1989). The design of browsing and berrypicking techniques for the online search interface. *Online Review, 13,* 407–424.

Belkin, N. J., Kantor, P., Cool, C., & Quatrain, R. (1994). Combining evidence for information retrieval. In D. K. Harman (Ed.), *Second Text Retrieval Conference (TREC-2), Gaithersburg, MD, USA, 31 Aug.–2 Sept., 1993* (pp. 35–44). Washington, DC: National Institute of Standards and Technology.

Bookstein, A. (1983). Outline of a general probabilistic retrieval model. *Journal of Documentation, 39,* 63–72.

Bookstein, A. (1985). Probability and fuzzy-set applications to information retrieval. *Annual Review of Information Science and Technology, 20,* 117–151.

Buckland, M. K., Butler, M. H., Norgard, B. A., & Plaunt, C. P. (1993). OASIS: Prototyping graphical interfaces to networked information. In *Integrating Technologies, Converging Professions (Proceedings of the 56th ASIS Annual Meeting, Columbus, OH, Oct. 24–28, 1993)* (pp. 204–210). Medford, NJ: Learned Information.

Cooper, W. S., Chen, A., & Gey, F. C. (1994a). Experiments in the probabilistic retrieval of full text documents In *Text Retrieval Conference (TREC-3) Draft Conference Papers.* Gaithersburg, MD: National Institute of Standards and Technology.

Cooper, W. S., Gey, F. C., & Chen, A. (1994b). Full text retrieval based on a probabilistic equation with coefficients fitted by logistic regression. In D. K. Harman (Ed.), *Second Text Retrieval Conference (TREC-2), Gaithersburg, MD, USA, 31 Aug.–2 Sept. 1993,* NIST-SP 500-215 (pp. 57–66). Washington, DC: National Institute of Standards and Technology.

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)* (pp. 198–210). New York: ACM.

Fox, E. A., Akscyn, R. M., Furuta, R. K., & Leggett, J. J. (Eds.). (1995). Digital libraries [Special issue]. *Communications of the ACM, 38*(4), 23–96.

Fox, E. A., France, R. K., Sahle, E., Daoud, A., & Cline, B. E. (1993). Development of a modern OPAC: From REVTOLC to MARIAN. In *SIGIR '93 (Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, June 27–July 1, 1993)* (pp. 248–259). New York: ACM.

Fox, E. A., & Koll, M. B. (1988). Practical enhanced Boolean retrieval: Experiences with the SMART and SIRE systems. *Information Processing & Management, 24*(3), 257–267.

Fox, E. A., & Shaw, J. A. (1994). Combination of multiple searches. In D. K. Harman (Ed.), *Second Text Retrieval Conference (TREC-2), Gaithersburg, MD, USA, 31 Aug.–2 Sept. 1993)* (pp. 243–252). Washington, DC: National Institute of Standards and Technology.

Fuhr, N. (1992). Probabilistic models in information retrieval. *Computer Journal, 35,* 243–55.

Goldfarb, C. F. (1990). *The SGML handbook.* New York: Oxford University Press.

Hildreth, C. R. (1989). OPAC research: Laying the groundwork for future OPAC design. In C. R. Hildreth (Ed.), *The online catalogue: Development and directions* (pp. 1–24). London: The Library Association.

Keen, E. M. (1992). Term position ranking: Some new test results. 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)* (pp. 66–76). New York: ACM.

Larson, R. R. (1991a). Between Scylla and Charybdis: Subject searching in the online catalog. *Advances in Librarianship, 15,* 175–236.

Larson, R. R. (1991b). Classification clustering, probabilistic information retrieval, and the online catalog. *Library Quarterly, 61,* 133–173.

Larson, R. R. (1991c). The decline of subject searching: Long-term trends and patterns of index use in an online catalog. *Journal of the American Society for Information Science, 42,* 197–215.

Larson, R. R. (1992). Evaluation of advanced retrieval techniques in an experimental online catalog. *Journal of the American Society for Information Science, 43,* 34–53.

Lynch, D. (1995). *Implementing explain.* Available: ftp://ftp.loc.gov/pub/z3950/articles/denis.ps

Matthews, J. R., Lawrence, G. S., & Ferguson, D. K. (1983). *Using online catalogs: A nationwide survey.* New York: Neal-Schuman Publishers.

Ousterhout, J. K. (1994). *Tcl and the Tk Toolkit.* Reading, MA: Addison-Wesley.

Porter, M., & Galpin, V. (1988). Relevance feedback in a public access catalogue for a research library: Muscat at the Scott Polar Research Institute. *Program, 22,* 1–20.

Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science, 41,* 288–297.

Turtle, H. R., & Croft, W. B. (1990). Inference networks for document retrieval. In J. Vidick (Ed.), *Proceedings of the 13th International Conference on Research and Development in Information Retrieval, Brussels, Belgium, 5–7 Sept. 1990* (pp. 1–24). New York: ACM.

Turtle, H. R., & Croft, W. B. (1992). A comparison of text retrieval models. *Computer Journal, 35,* 279–290.

Walker, S. (1987). OKAPI: Evaluating and enhancing an experimental online catalog. *Library Trends, 35,* 631–645.

Walker, S. (1989). The Okapi online catalogue research projects. In C. R. Hildreth (Ed.), *The online catalogue: Development and directions* (pp. 84–106). London: The Library Association.