Search Interface for Z39.50 Compliant Online Catalogs Over The Internet

Danny C.C. POO
Teck Kang TOH
School of Computing
National University of Singapore
Lower Kent Ridge Road, Singapore 119260
d poo@comp.nus.edu.sg
tohteckk@comp.nus.edu.sg

Christopher S.G. KHOO
Centre for Advanced Information Systems
School of Applied Science
Nanyang Technological University
Singapore 639798
assgkhoo@ntu.edu.sg

Abstract

Reference Librarians play an important role in library premises. They help users in getting the most out of their searches and getting what they want more effectively. Such services are usually not available over the Internet for practical reasons. However, with a search interface playing the role of a Reference Librarian, similar services in helping users conduct more effective searches of the catalogs can be provided over the Internet.

This paper discusses the problems users generally encounter in using online catalogs and discuss the design of a web-based search interface for online catalog systems searches over the internet.

1. Introduction

Increasing numbers of online library catalogs are becoming accessible on the World Wide Web. However, most of the Web interfaces are little better than the traditional text-based online catalog interfaces [5]. Studies have shown that users have problems using the online catalogs effectively.

Reference Librarians are the ones that users usually turn to for help. This is because Reference Librarians possess searching skills and knowledge about catalog systems that users do not have. Help from Reference Librarians is not available away from library premises, e.g., when users search catalogs over the Internet.

This paper proposes the use of a search interface that provides users with some assistance in their online catalog searches over the Internet. The search interface is Z39.50 compliant; this means that it can search across multiple online catalog systems.

This paper is organized as follows:

• We begin with the problems users generally encounter in using online catalogs and discuss how we hope to address the problems with the search interface.
• We describe the design of the web-based search interface for online catalog systems.
• We discuss the search strategies used by the search interface in a search session on the catalog systems.

The search interface is currently being implemented and experiments will be conducted to test the effectiveness of the search interface against the conventional Z39.50 search systems and human expert librarians.

2. Problems with Online Catalogs

Online catalog systems have existed for many years but there has been little improvement in the design of the systems in terms of helping the users to search for information effectively. Markey [16] found that users lack the knowledge to do the following:

• match their terms with those indexed in the online catalog.
• identify terms broader or narrower than their topic of interest.
• increase the search results when too little or nothing is retrieved.
• reduce the search results when too much is retrieved.
• make effective use of the Library of Congress Subject Headings used to index the records.

Borgman [2] noted that online catalog systems "were designed for highly skilled searchers, usually librarians, who used them frequently, not for novices or for end-users doing their own searching." She found that effective online catalog searching required the following types of knowledge:
• Conceptual knowledge of the information retrieval process – required for translating an information need into a searchable query.
• Semantic knowledge of how to implement a query in a given system – required for knowing when and how a system feature is to be used.
• Technical skills in formulating the search statement in the language and syntax required by the system.

In addition, Hildreth [13] pointed out that the online catalog systems place the burden on the user to reformulate and re-enter searches until satisfactory results are obtained. There is not enough help in the system to assist users faced with these difficulties. Recent improvements to online catalogs, Borgman [2] noted, have been in surface features rather than in core functionality.

Studies conducted by Cousins [6], Dalrymple [8], Ensor [12], Lancaster [15], Markey [16] have found deficiencies in the present-day online catalog systems and identified the problems users have. For example, Cousins [6] analyzed the types of subject queries that users brought to two online catalog systems. She found that many subject queries were not expressed at the level of specificity that was appropriate or suitable for searching the system. She concluded that online catalog systems should provide more information about the document content (e.g. content pages), facilities for browsing the thesaurus and the classification scheme, facilities for browsing records arranged by class number, ranked display of search results, help with query formulation, and relevance feedback.

Attempts have been made to develop expert system interfaces to help users search online catalogs more effectively. Drabenstott and her colleagues [9][10][11] have developed a prototype online catalog system that uses search trees or decision trees to represent how experienced librarians select a search strategy and formulate a search statement. The decision tree is represented as a flowchart. In an evaluation using a small database of 15,000 records, she found that the search trees were more effective than selecting a search strategy at random. However, decision trees are relatively rigid and the selection process goes in one direction. Adding new search strategies to the model may not be easy because it necessitates changes to many parts of the model. It may not be easy to customize the model for particular users and user groups. Processing of the user query can only go forward following the direction indicated by the flow chart, i.e. it cannot backtrack unless explicitly indicated in the flowchart and cannot test a few search strategies before selecting one (i.e. it cannot search backwards).

Chen's [3][4] system was developed as part of his dissertation, and was an ambitious system with many search strategies and knowledge bases. The evaluation was carried out with a very small database of 300 records. It is not clear whether similar or better results can be obtained with a much smaller and simpler system.

Users’ expectation of catalog systems has increased with the ease in communication made possible through the Internet. Users now want to be able to search across multiple catalog systems. To search multiple catalogs, users have to learn and use many different search systems with different interface designs and search capabilities. This exacerbates the situation.

3. Features of a Z39.50 Compliant Search Interface

The proposed search interface has been designed with the following features:
• Z39.50 Standard Compliant [1].
• The utilization of search strategies used by Reference Librarians.
• A production rule-based expert system that makes it easy to add and modify search strategies.
• A simple and easy to use search interface.
• A web-based interface for connecting to the Internet.

3.1. Z39.50 Compliant Search Interface

One problem faced by users is the need to learn a different search system for each online catalog. This can be resolved by implementing the search interface based on a widely-used protocol — the Z39.50 Information Retrieval Protocol.

Z39.50 is an American National Standard that was approved in 1988 by the National Information Standards Organization (NISO). The standard provides a uniform procedure for client computers to query information resources such as server computers hosting a library catalog.

The protocol allows the search interface to access any Z39.50 compliant online library catalog system. Thus, users of the search interface are able to access a wide variety of information resources. Also, the result sets of Z39.50 queries are maintained at the server, and they can be retrieved piece-meal as and when necessary. This feature is especially useful when the result set is large and returning all of it at one go may not be effective due to the limited bandwidth of current networks.

3.2. Utilization of Search Strategies Used by Reference Librarians

Current search systems do not utilize the various search strategies of Reference Librarians to help users in their search. Thus, the search interface is designed to take
full advantage of the strategies used by Reference Librarians to help users search more effectively.

### 3.3. Production Rule-based Expert System

When designing the search interface, we noted that search strategies would be selected based on some selection rules. The search interface was thus designed as a Production Rule-based Expert System as this is the most natural and straightforward way to implement the selection rules and strategies used in the search process. Modification of search strategies can also be easily carried out with this approach.

### 3.4. Simple and Easy to Use

The objective of the search interface is to help users search more effectively. A simple and easy-to-use search interface would certainly help.

### 3.5. A Web-based User Interface

With the widespread use of Internet, a web-based user interface is inevitable. By designing the search interface to be web-enabled, easy access to the search interface is possible.

### 4. Design of the Search Interface

The design of the search interface is shown in Figure 1. It consists of the following modules:

- Graphical User Interface Module
- Knowledge Module
- Control Module
- Expression Module
- Z39.50 Interface Module
- Subject Module
- OCLC (Online Computer Library Center) Z39.50 Client API for Java.

#### 4.1. Z39.50 Interface to Online Library Catalog

Development effort of the search interface can be greatly reduced with the OCLC Z39.50 Client API for Java. The Z39.50 Interface Module is designed as a wrapper module that interacts with the OCLC Z39.50 Client API, to provide functionality for connecting, searching and retrieving information from the various online catalog systems.

#### 4.2. Control Module

JESS [14], a Java Expert System Shell, is used to develop the expert system aspect of the search interface. It is a clone of the popular expert system shell CLIPS [7] but rewritten entirely in Java. With JESS, we are able to integrate other Java code into the shell easily and provide the search interface with the ability to reason.

At the heart of the expert system is the Control Module. It is designed to control and implement the various functions of the system using a script. It has the following integrated components:

- A Knowledge Base of search strategies specified in terms of rules in the form of JESS script.
- A Fact Base which stores all the intermediate search results and information needed to select the next search strategy in the JESS engine. A new strategy is selected by evaluating the rules (from the Knowledge Base) against the facts (in the Fact Base).
- An Explanation Facility for explaining why and how certain strategies were chosen.
- A Session Log that keeps a permanent record of all the search results and strategies used. The Log is used to analyze the performance of the search interface.

---

**Figure 1: Design of the search interface**

#### 4.3. Knowledge Module

The Knowledge Module contains wrapper functions for integrating the JESS scripts of the Control Module with the other modules of the search interface. This is necessary since the other modules such as the Expression
Module, Subject Module and the Z39.50 Interface Module are in Java and they have to interact with the Control Module which selects and executes the appropriate search and reformulation strategies based on the rules in the Knowledge Base.

4.4. Graphical User Interface Module

The Graphical User Interface Module is designed to provide users with greater ease in the use of the search interface. Graphical functions such as the Input Dialogs and Feedback Dialogs are wrapped as JESS functions by the Knowledge Module.

4.5. Expression Module

The Expression Module removes suffixes (using Porter’s algorithm [18] from the user’s search query and provides other functions for manipulating a search expression. It has functions for removing “stopwords”, converting a search expression to reverse polish notation, removing AND and OR operators from an expression, converting stopwords to the AND operator, etc. The various functions can be called by a search strategy.

4.6. Subject Module

The Subject Module handles the reformulation of the initial search query into a new search query by analyzing the terms found in the records retrieved by the initial search. The keywords and subject headings found in the records returned by the initial search are scored using a formula and ranked. A new query is then reformulated with the keyword or subject heading that has the highest ranking. The reformulation strategy is based on the assumption that records containing the keyword or subject heading with the highest ranking are more likely to be relevant to the user’s search objectives.

5. Search Strategies

Two types of search strategies are used in the search interface:
1. Initial Search Strategies.
2. Reformulation Strategies.

5.1. Initial Search Strategies

These strategies are used to convert the user’s natural language query statement to a formal Boolean search statement before sending to the library catalog system. “Expert systems in library reference service” and “A study on computer literacy and use by teachers in primary schools” are examples of natural language query statements. Most library systems still require the query statement to be expressed using Boolean operators.

Two types of initial search strategies are used:
- Keyword search in all fields.
- Subject Headings search.

5.1.1. Keyword Search in All Fields

Each strategy adopted by the search interface can be described as a procedure of actions. This strategy has the following sequence of actions:
1. User enters query, e.g. “Expert systems in library reference service”.
2. Remove stopwords and punctuation. This reduces the query to “expert systems library reference service”.
3. Insert Boolean AND between the keywords. The search query is now “expert AND systems AND library AND reference AND service”.
4. Stem the words and add a truncation sign at the end of every word. The result is “expert? AND system? AND librar? AND refer? AND servic?”.
5. Formulate a search statement to carry out a keyword search in all fields. The search statement sent to the catalog system is “find expert? AND system? AND librar? AND refer? AND servic?”.

This strategy is a simple one that basically removes stop words, stems words, inserts the boolean “AND” between all words, and carries out a keyword search in all searchable fields in the catalog system.

5.1.2. Subject Headings Search

This strategy makes use of the Library of Congress subject headings in bibliographic records to gather appropriate records for users to decide. It selects the shortest subject heading that contains the highest number of query words. Non-preferred terms in the Library of Congress Subject Headings list are selected in the same way but are replaced with the valid subject heading when the stemmed query statement is constructed. Up to 2 subject headings are selected and linked with the boolean “AND”. Query words that are not part of the selected subject headings are combined with the latter using “AND” to form the final search statement. This strategy has the following steps:
1. User enters query.
2. Remove stopwords and punctuation, and stem the words.
3. For each stemmed keyword in the query, identify the subject heading-subdivisions as well as non-preferred terms beginning with that keyword.

4. Assign a score to each subject heading equal to the number of query words that it contains.

5. Identify the subject heading with the highest score. If there is only one with the highest score, then select that heading. If there is more than one with the highest score, then select the heading(s) containing the smallest number of words (i.e., the shortest heading). We shall refer to the headings selected at this point as the primary subject headings.

6. For each primary subject heading, identify the query words (if any) not found in the heading. If there are one or more query words not found in the heading, then identify a second subject heading containing the highest number of the remaining query words (repeating steps 3-5 above).

7. Each primary subject heading is now paired with none, one, or more secondary subject headings. Link the primary and secondary subject headings with Boolean AND.

8. Combine the pairs of subject headings with Boolean OR.

9. Replace the non-preferred terms with the valid subject headings, and formulate a search statement to submit to the library system.

5.2. Reformulation Strategies

Reformulation strategies are used after the results of the initial search have been obtained and examined by the user. Only the first twenty titles retrieved from the catalog system will be displayed. The user scans the titles, and indicates which titles are relevant to his query.

Based on the user's relevance feedback, the strategy may modify the previous search statement or generate a brand new query statement.

Three types of reformulation strategies are used:

- Broadening Strategies.
- Narrowing Strategies.
- Relevance Feedback Strategies.

5.2.1. Broadening Strategies

A broadening strategy modifies the previous search statement to make it less constrained. This strategy will generally retrieve more records, and is appropriate when no record is retrieved or when most of the records retrieved are relevant and the user wants to get even more records.

Broadening strategies used by the search interface are:

- Strategy B1: Convert adjacency operators to boolean ANDs
- Strategy B2: Search each query word individually to identify those not found in the catalog system. Remove such words from the search statement.
- Strategy B3: Select every combination of 3 query words. Link the words within a combination with boolean AND. Link the combinations with OR.
- Strategy B4: Select every combination of 2 query words, and OR the combinations.
- Strategy B5: Convert all the AND operators to "OR".
- Strategy B6: Prompt user to enter synonyms and related words for each query word.
- Strategy B7: Use a broader subject heading.
- Strategy B8: Use stronger stemming.

5.2.2. Narrowing Strategies

A narrowing strategy aims to reduce the number of records retrieved from the catalog system. The set of records retrieved will be a subset of the earlier result set. A narrowing strategy is appropriate when too many records are retrieved, and the user wants to reduce the set to those records that are more likely to be relevant.

Narrowing strategies used by the search interface are:

- Strategy N1: Replace an OR operator in the search statement with AND, and execute the search. Do this for each OR operator in turn. Combine all the search sets using OR.
- Strategy N2: Convert an AND operator to an adjacency operator, and execute the search. Do this for each AND operator in turn. Combine all the search sets using OR.
- Strategy N3: Ask the user for additional keywords to AND to the search statement.

5.2.3. Relevance Feedback Strategies

A relevance feedback strategy is based on the assumption that when a keyword or subject heading occurs in most of the relevant records but only a few non-relevant records (as indicated by the user), that keyword or subject heading is likely to retrieve other relevant records. The strategy analyzes the content of the records retrieved and attempts to identify these keywords and subject headings.

A list of terms (keywords and subject headings) is first compiled from the twenty records displayed to user. A set of statistics is derived for each of the terms:

- number of relevant records containing the term
These statistics are also calculated for each pair of terms (pair of keywords, pair of subject headings, or keyword plus subject heading). A score is then generated for each term and pair of term using one of the following formulae:
1. number of relevant records minus number of non-relevant records
2. percentage of relevant records minus percentage of non-relevant records
3. number of relevant records divided by number of non-relevant records
4. percentage of relevant records divided by percentage of non-relevant records
5. number of relevant records
6. percentage of relevant records.

Experiments will be carried out to investigate which of these formulae work best to identify the best term or combination of terms to use.

5.2.4. Rules for Selecting a Reformulation Strategy

The search interface makes use of the decision table in Table 1 to select a reformulation strategy.

<table>
<thead>
<tr>
<th>Condition 1</th>
<th>Condition 2</th>
<th>Strategy Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 record retrieved</td>
<td>Query contains only 1 word (after removing stopwords)</td>
<td>Broadening strategy 6: Prompt user to enter synonyms and related words</td>
</tr>
<tr>
<td>20 or fewer records retrieved (all displayed to the user)</td>
<td>All records are relevant</td>
<td>Relevance feedback strategy</td>
</tr>
<tr>
<td>More than 20 records retrieved</td>
<td>All records are relevant</td>
<td>Display 20 more records</td>
</tr>
<tr>
<td></td>
<td>All records are not relevant</td>
<td>Narrowing strategy</td>
</tr>
<tr>
<td></td>
<td>Some records are relevant, some are not</td>
<td>Relevance feedback strategy</td>
</tr>
</tbody>
</table>

6. Conclusion

In this paper, we described the problems with conventional search systems in performing searches. In addition, we have described the design of an expert intermediary search interface that we believe will help the users in performing searches.

The search interface is currently being developed as a prototype to test the various search strategies.

Experiments will be conducted to test the effectiveness of the design of the search interface. Two phases of experiments have been scheduled. The first phase experiments will test the effectiveness of the initial search strategies, and the second phase experiments will test the effectiveness of the reformulation strategies.

The search interface will also be tested with sample queries against a conventional Z39.50 search system and a human expert librarian. The results of the experiments will be reported in a separate paper.

REFERENCES


