“Quality Metrics” Project: Second Interim Report

Aaron Krowne, Rohit Chopra, Urvashi Gadi,
Martin Halbert (Emory University)
Vikram Raj, Edward A. Fox (Virginia Tech)

November, 2005

Contents

1 Introduction

1.1 Purpose of This Report

This document is the second interim report of the "Quality Metrics" IMLS NLG project by Virginia Tech and Emory University, formally titled *Study of User Quality Metrics for Metasearch Retrieval Ranking*. It serves as a complete update on the status of our activities since the first interim report (July 2005). These activities include management of the project as well as research and development progress.

1.2 Project Background

Digital library search systems have not evolved to keep up with growing user expectations and the metadata-rich nature of digital library objects. These search systems, though an improvement from arcane, classical OPACs, behave clumsily in the face of heterogeneity and radically different values for important metadata attributes. The increasing prevalence of metasearch, a scenario whereby disparate and generally heterogeneous objects are searched together in one interface, has exacerbated the problem.

The reason for this situation is that search engines have come out of the field of information retrieval (IR), which has recently been focused on solving the web search problem. While digital libraries have benefitted from recent advances in information retrieval, largely spurred to solve the web search problem, the web search and digital library search scenarios are quite different. Aside from scale differences and different amounts and kinds of item interlinkage, the web largely lacks metadata.

This is not a minor distinction. As an example of how this manifests, digital libraries have the opportunity to distinguish between item-level and collection-level records in determining how to present retrieval results. Search engines developed for web-based materials do not address this issue of record type (which is also an issue of *granularity*). Some search engines adapted for digital libraries have attempted to differentiate results along such granularity boundaries, but their approaches have not been formally tested with users. A similar problem also manifests in treating results from various subcollections (e.g. separate library content databases, metadata records harvested from other digital libraries) and items culled from the web vs. "native" digital library records.

Further, these and other attributes often also bear on notions of *quality*, which deeply influences the organization of retrieval results. In the standard web search situation (Brin and Page, 1998), hyperlink data is used to make inferences about quality which allow ranking to be vastly improved over purely content-based matching methods. In digital libraries, we could extend the gathering of such quality information to attributes pertaining to vettedness of records, rating, view popularity, logged activation from search results lists, aggregation in path or lesson plan objects, and much more. Thus, there is both a great need and opportunity to intelligently make use of digital library metadata in retrieval results presentation.

The goals of this project are (1) to discover the best way to present digital library retrieval results by digging down to the user expectations level, (2) apply these findings to a working prototype system, and then (3) evaluate the effectiveness of these systems relative to standard or typically available alternatives.

We will refer to the new search system we are developing to integrate and expose quality metrics as "QM-search."

2 Emory/VT Meetings

We have had two meetings since the last interim report between the team components at Virginia Tech and Emory University. The first, on October 14 2005, was between Ed Fox and the Emory team (at Emory). The second was between Vikram Raj and and the Emory team (also at Emory).

During the first meeting, the Emory team briefed Ed Fox on work to date, chiefly on the coordination of the focus groups. There was much discussion of the plans for conducting the focus groups, and Ed Fox provided much useful commentary on these plans. His feedback subsequently led to considerable revisions. We also discussed data sources for quality indicator information, which we will need to use for modelling and experimentation. Finally, we discussed our progress on developing QM-search, and the need to acquire analysis systems to "plug" into QM-search to generate quality metrics (key code having been written by Wensi Xi and Seonho Kim at Virginia Tech).

In the second meeting, Vikram Raj visited Emory. A large component of the meeting was bringing Vikram up to speed on our activities with the focus groups. By this time, we had run a few sessions and had learned some things about modelling the quality needs in metasearch retrieval, as well as learned many things about running user studies. We communicated these things to Virkam, and discussed the implications for his running experiments later at Virginia Tech.

We also got together and viewed one of the focus group session recordings, which was an extremely enlightening activity. Vikram told us this gave him a very good feel for what we had done and what the project was about in general. Even those of us at Emory who had been present at the sessions realized that one could gleam numerous additional insights from re-watching the recorded session.

Finally, we discussed experimental design and logistics of running the experiments at Virginia Tech. We came up with some basic sketches of the experiment . One important aspect of this which we discussed was selection of the study participants, balancing issues of cost and availability with impact and generalizability of the results.

Due to these meetings, on top of our e-mail and phone communications, we feel that the Virginia Tech and Emory groups are remaining well-coordinated.

3 User Studies

In this chapter we discuss our progress on user studies to date. The user studies that have been partially conducted so far are the focus groups at Emory. We split the discussion of these into three sections. The first covers our execution of the focus groups, discussing how we are running them, and how we have been altering our plans in response to what we've encountered. The second discusses our major findings from these focus groups. The third discusses how the QM-search mock-ups, which are integral to the focus groups, have been formulated and how they have changed. Finally we discuss planning for the capstone evaluation of the focus groups and the data generated by them.

3.1 Execution of Focus Groups

During this project interim, Rohit Chopra wrote and submitted an IRB proposal for the focus groups. This proposal was subsequently approved. Chopra is the lead coordinator of the focus groups, with Aaron Krowne, Katherine Skinner, Urvashi Gadi, and Martin Halbert participating in a supporting capacity.

So far we have held six focus groups. Each of these is a one 1:15 "lunch session," for which we provide food at no cost to the participants. We feel this arrangement minimizes inconvenience on prospective attendees, making them more likely to attend. Participants are introduct to the investigators and to each other. We have included faculty, library staff, and graduate students in the focus groups (done separately so as to control for inter-group influences). So far we have covered grad students in the humanities and faculty in the humanities.

The "pre-session" is a 15-minute period reserved entirely for lunch and introductions. The sessions then begin in full with a short (5-7 minutes) IRB informed consent disclosure and general discussion of the IRB process. Next, we introduce the topic of the investigation, giving a lay description of the quality metrics problem for metasearch retrieval systems (10-12 minutes). We are currently formatting this as a hybrid lecture/scenario; when possibly trying to couch the explanation in a step-by-step sequence of events a typical scholarly user might go through. Participants are allowed to interrupt to ask questions at any time, which is an important part of using the time efficiently and ensuring that the participants can take every opportunity to understand what they are seeing and being told.

The last 30-40 minutes of the session is semi-structured. We begin this part by asking users broad questions (sometimes informed by questions they have already asked) to get them thinking, talking, and discussing. Throughout the semi-structured portion, we ask questions based on issues, comments, and observations that arise. Thus, this portion is very unpredictable, but is also full of many nuanced insights.

Some examples of high-level questions we have asked to foster discussion are:

- Think of your own scholarly inquiries for which you used search systems, and think about what worked well and what did not work well about these. Comment, or show us.
- When you look at search results, what information displayed forms the best indicators to you of the quality or pertinance of those results?
- What kind of quality indicator information would you like to see used in metsearch systems?
- Which quality indicators would you like to see the results broken down, filtered, or sorted by?
- Do you ever use complex interfaces/advanced searches?
- Do you move from simple to complex interfaces? When? Why?
- What is an interface that is too complex? Too simple?
- Looking at an interface like A9 (or other systems you use frequently), what would you change?

Throughout the sessions, when using our mock-ups and in utilizing scenarios in general, we structure them around queries such as "postmodernism," "globalization," or "civil rights movement." Basically the objective here is to include everyone in the focus group by utilizing topics they can all understand and relate to. We think inquiry scenarios like these afford a scholarly (and in specific, humanities) bend, without being overly specific and arcane.

The physical setup of the focus groups consists of the electronic distance learning classroom in the ECIT center of Emory's Woodruff library. This room is equipped with a two-display rear projector array in front (which is connected to a standard computer) and a semicircle seating area facing this. We have set up two cameras in the room to record the sessions, one of which is placed facing the screen, and one of which faces the participants. These recordings are immediately digitized for convenience and permanence. Participants may eat at the main table, so that they do not need to ever face away from the screen and miss out on the discussion.

There are no separate computers for the participants, but we provide a wireless keyboard and mouse which can be passed around to allow interaction with metasearch sites. Participants often took advantage of this, or simply told us what to do on screen to demonstrate things or investigate various possibilities.

This setup has the advantage that everyone gets to see the same thing and provide feedback on it, thus, everyone is "on the same page."

We have discovered numerous things that "work well" in running these focus groups. For one, mock-ups are very useful; we are getting extensive feedback about them, and users are able to "grasp" them. We are also finding that bringing live systems into the discussions is very useful. This tends to "latch on" to the participants' existing experience, and at the same time, we can leverage this to explain how various aspects of these systems are and need to be generalized by our quality metrics framework. We have had especially good luck with "inspiration" systems like A9 (covered in the first interim report); even if participants have never used these systems, we can get a lot of good feedback from them after taking a few seconds to demonstrate them, after which the participants typically thank us for showing them something so useful.

3.1.1 Field of Study Aspects

Interfaces used in the last two focus groups (and which will be used in the next 3 focus groups this semester) have had southern studies oriented content, e.g. representation of "civil rights movements" records. All faculty members participating in these groups have not necessarily been / will not be from southern studies. However, we have had / will have several faculty with a primary or secondary southern studies focus.

In the remaining focus groups this semester we are largely including humanities faculty, but at least one focus group will be a mix of science and humanities faculty. Next semester we aim to get more grad student and faculty participants from the sciences.

3.1.2 Changes To Focus Group Script

As we progress through the focus groups, we have been striving to improve our "script" for how they are conducted. Prime objectives are:

- Minimize time where we are "lecturing."
- Maximize how comprehensible the concepts we are trying to convey are to the participants.
- Capture unexpected and nuanced insights from the participants.
- Maximize the feedback we get from all of the participants who attend.

Some key changes we have been striving to make to further these objectives are:

- Using scenarios to introduce QM-search, both to shorten the introduction "lecture" as well as to increase its impact and pertinance to the participants.
- Opening all portions of delivery up to question and comment.

- Minimizing abstract information at the beginning, instead showing off our model as "instances" and only abstracting later when the participants essentially "stumble upon" the abstractions themselves.
- Revising the mock-ups.
- Heavier use of live metasearch systems (some of which we've integrated after past participants showed them to us).

We feel that the results of making these changes have been positive, though by no means are we done making improvements.

3.2 Major Findings to Date

We have performed no comprehensive review of the focus groups as of yet (especially since we are not done conducting them). However, we have noted a variety of significant points in our debriefing discussions. Some of the major ones are highlighted below:

- Users have vastly different needs in terms of quality indicators (which we predicted). However, the need varies much more widely based on *discipline and research topic* than based on user *category* (i.e., undergrad, grad student, professor, staff/faculty).
- We have learned some new indicators we could implement (though not necessarily easily) which some scholars would find interesting, for example, whether a published article comes from a secular versus a theological source.
- We found that some users have an emphasis on "deep" indicators such as academic credibility, but some do not and are more interested in record type and fitness to the inquiry. Some of this stems from the fact that there exists no good prevailing quality indicators in certain new or extremely interdisciplinary fields.
- Users definitely see filtering as a transparent part of the retrieval process—they want filtering operations built into the retrieval interface.
- Filtering should be made "transparent" as much as possible (i.e., combine classification-based filtering transparently with the search process, a-la UNC's library web site).
- Users don't want "modality" and a lot of context switches. This, in fact, is at the root of a lot of complaints with existing metasearch interfaces.
- Participants often switch to requiring a different set of quality metrics in the process of an inquiry. This suggests some degree of customization or completely re-organization of the quality metrics being used is needed. It

also suggests that users are ok with context switches that *they* initiate, but (combined with the above observation) are less tolerate of ones that are overly driven by arbitrary aspects of the library or the data itself.

- Participants are very interested in having accessibility qualities made more apparent (whether a resource has been digitized or whether it is just catalogued, where it is available physically, if it is available to them, what the cost and openness are, and so forth).

In response to some insights, we have made changes in the design and implementation of QM-search. For example, we now have more focus on building "link-type" information into QM-search (e.g. classifications, containment, ref link, etc.), which will allow more navigation and filtering to be performed "superimposed" over collections. In general, we have more of an emphasis on filtering, which is now a section of the organization specification which is interpreted by the QM-search system.

We still think more focus is needed on the evolution of retrieval sets and their presentation through time. In other words, as the user conducts filtering and re-organizing operations, this evolving process must be simple and intuitive, yet easily implementable in our framework and scalable.

3.3 Evolution of Mock-Ups

The mock-ups we used in the focus groups evolved considerably over the sessions we have had so far. Aside from optimization of the information layout and style, many changes were in response to feedback from the focus group participants.

In the figures that follow (3.1, 3.2, 3.3, and 3.4), we show an evolution of the "two- and three-dimensional" hypothetical QM-search results screens.

3.4 Planning for Evaluation of Focus Groups

The focus groups, as a source of data in the context of a research project, must be evaluated. Of course, some of this is being done continuously: it is simply the sum of observations and insights stemming directly from the focus group sessions themselves, and our debriefings of them (and discussions at other Quality Metrics meetings).

In addition to this we plan, at the end of the focus group process, to view the recordings of all the sessions, then combine new observations with our notes to produce summary observations. This will be a combination of "testing" our model of the problem space against the theoretical model we have constructed, as well as providing direction for the remainder of the project and material to form wider recommendations for the digital library community. Rohit Chopra will lead on this evaluation phase, and there will be a report by February 2006.

On top of this, we are considering the possibilty of more precise forms of evaluation of the focus groups, straddling the qualitative/quantitative dichotomy.

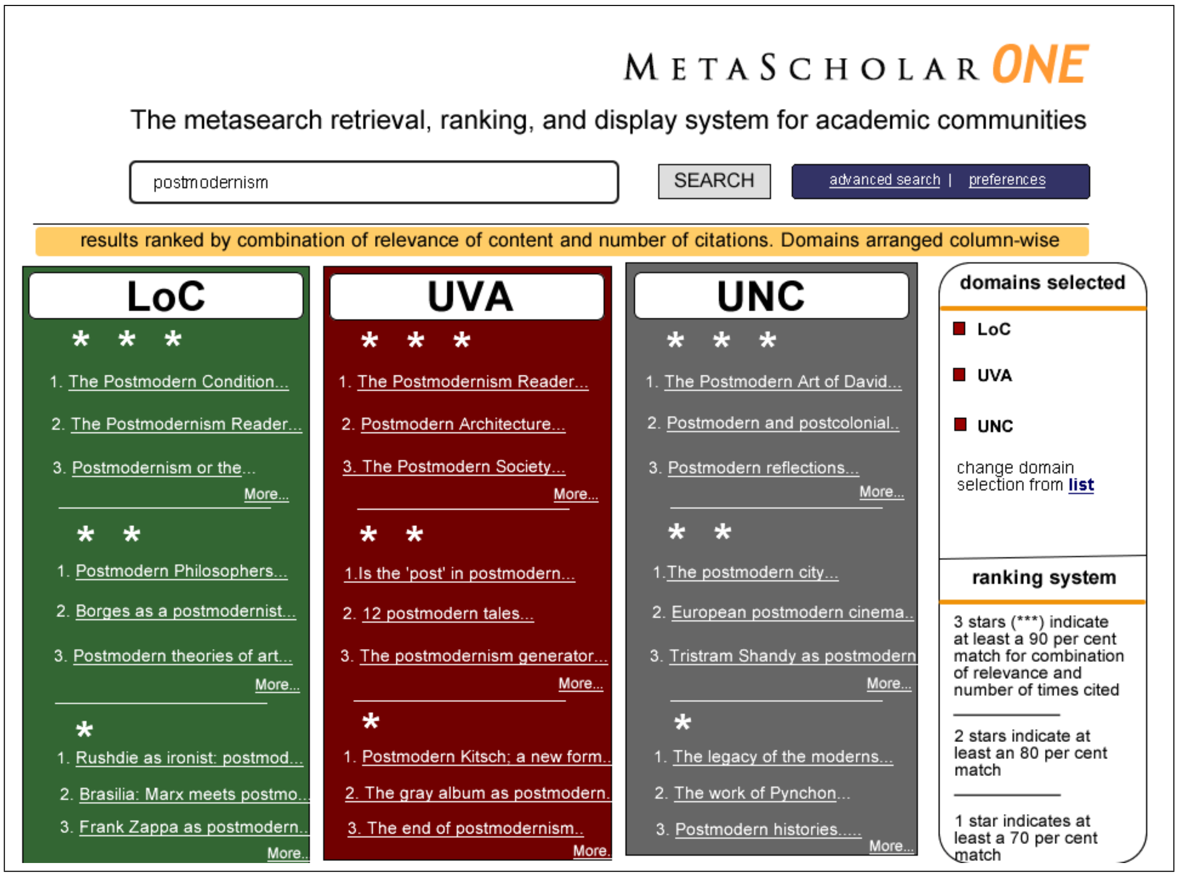

Figure 3.1: September 5. A "three-dimensional" search, with dimensions "domain," starred sections proportional to extent cited, and finally linear ranking by content-query similarity.

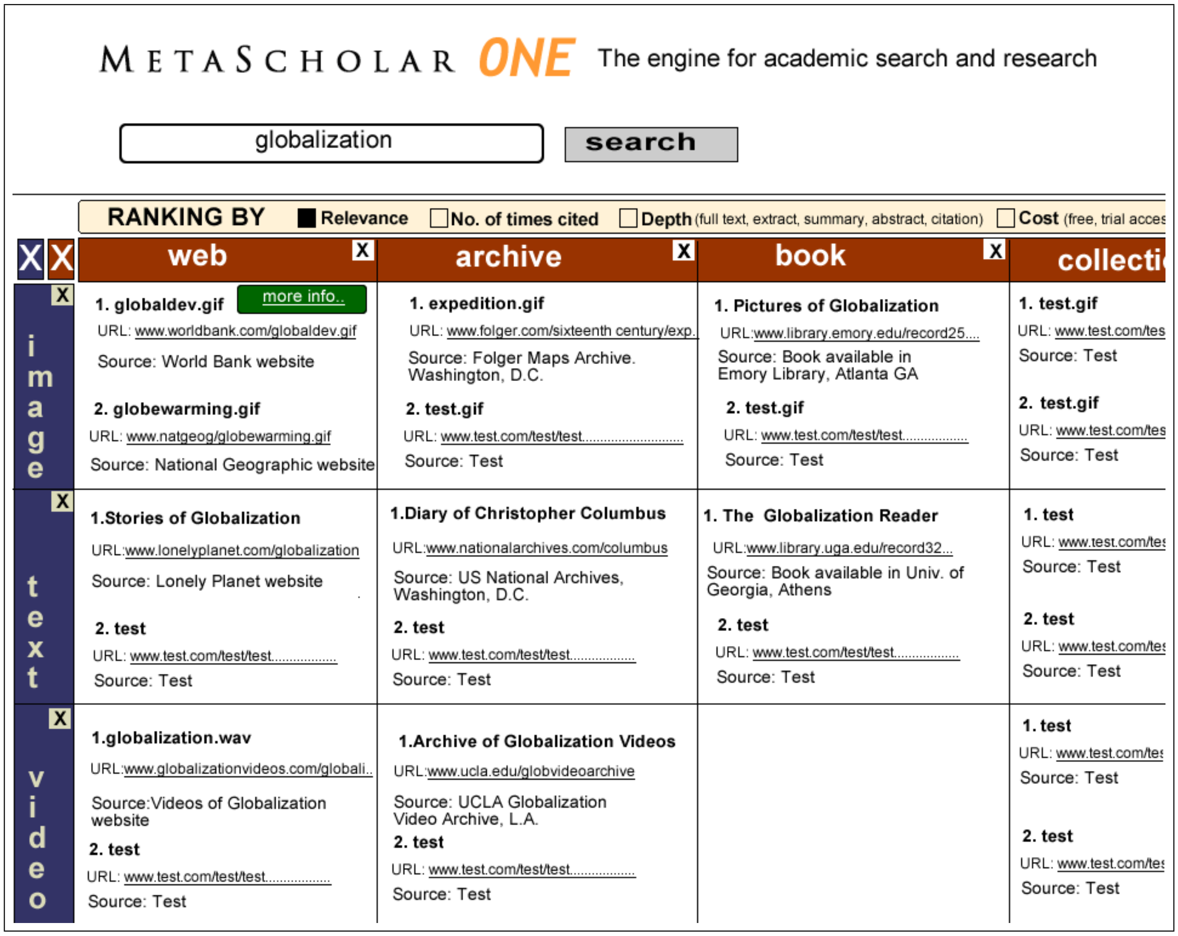

Figure 3.2: October 10. Another "3d" search, with dimensions of source *type*, media type, and finally content-query similarity.

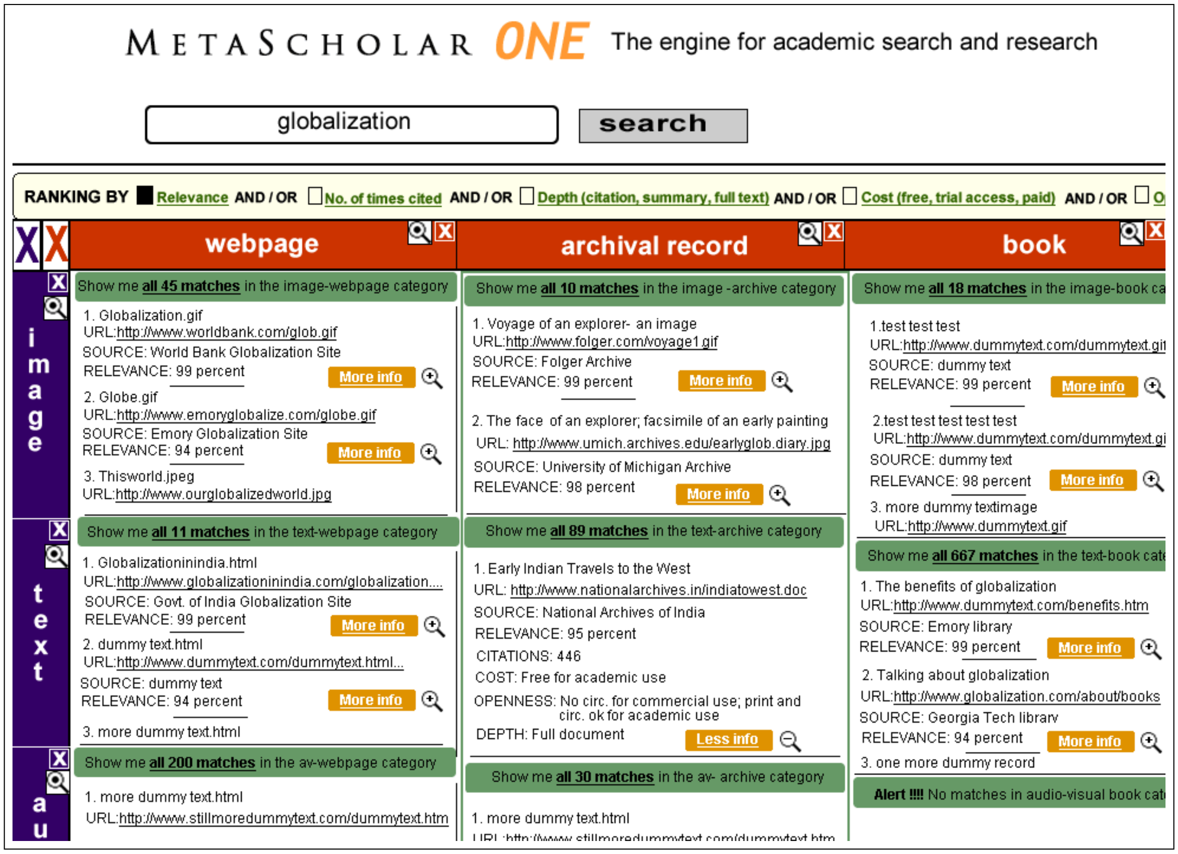

Figure 3.3: October 24. Version of the previous search, but with graphical changes and more descriptive metadata. Notice that we have added a "magnifying glass" icon so that one can "zoom in" (select only) particular columns or rows.

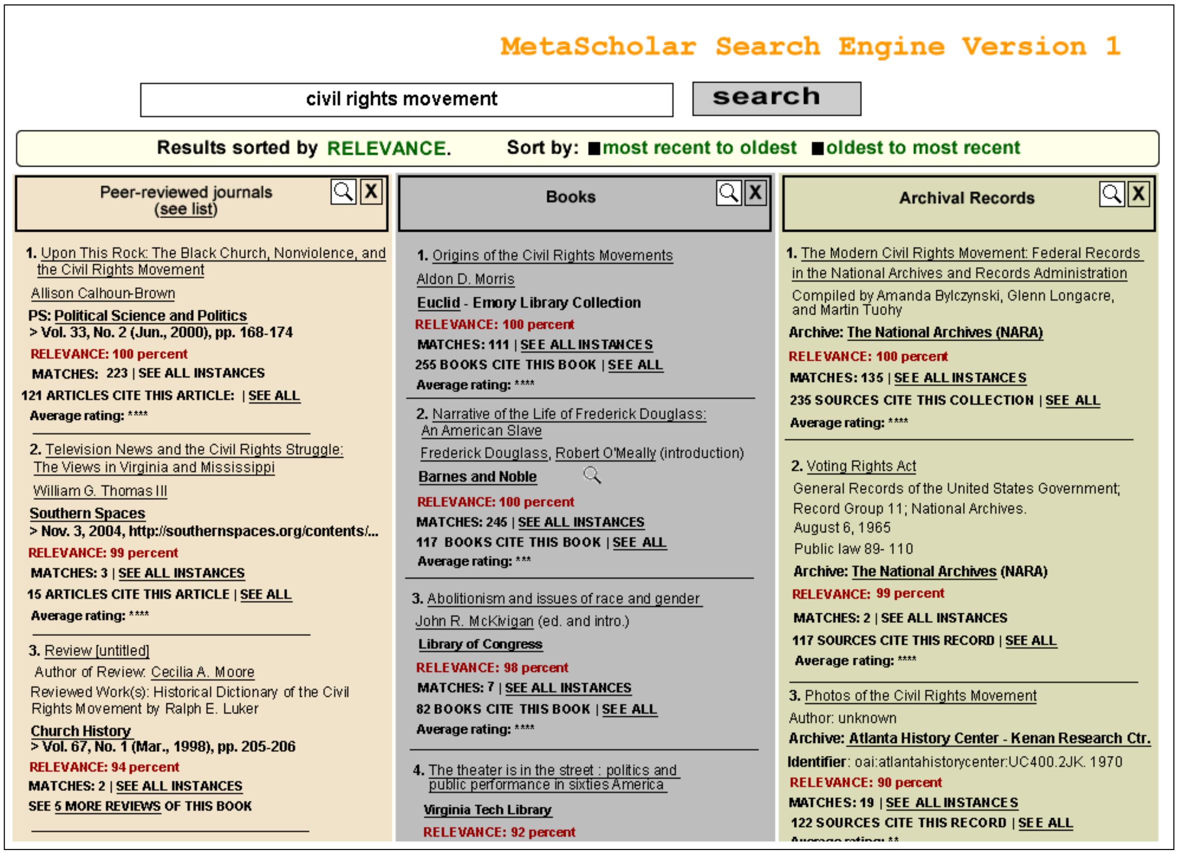

Figure 3.4: November 11. A "2D" display, which we moved to in order to lower the initial complexity of what we were showing the participants. Now, the linear rank (top to bottom) is a combination of content-query similarity, average rating, and citedness.

The way we would do this is develop (from colloquial observation of focus group sessions) a set of meaningful indicators which we feel are present in the interactions. Then, while re-watching the videos, we would keep track of when these indicators occur, for tabulation and reporting at the end.

Some kinds of indicators we could keep track of could be:

- Users liked/didn't like various metasearch systems (judging by overt statements).
- Count of "problems" or "good aspects" of existing metasearch systems.
- Which "dimensions" of the mock-ups users liked or didn't like, or understood/didn't understand.
- Which quality indicators the participants said were useful or not useful to them.

Caveats are of course that the more indicators we trawl from the recorded sessions, the longer this analysis will take (or, equivalently, the more shallow and inaccurate it will become). We plan to generate many more such indicators, and select a set of most promising candidates that are tractible to tabulate.

4 Experimental Studies Planning

Primarily at the Emory meeting with Vikram Raj, we formed some basic sketches of experimental design for the Virginia Tech experiments with the QM-search system. These sketches break down into two scenarios: the known-item search, and the open-discovery search. We feel that these two inquiry scenarios, both of which utilize metasearch systems, necessitate very different experimental evaluations. It also became clear that we would have to go far beyond the typical information retrieval metrics of precision and recall.

The sketches below will be developed further, improved upon, and vetted by the Virginia Tech team, in collaboration with expert colleagues in the wider digital library community and other members of the digital library research lab at Virginia Tech. A common element is that the experiments will be testing QM-search against one or more "representative" metasearch systems which already exist. We have not yet determined the selection of these systems. They could, in fact, be adaptions of off-the-shelf search systems (like Lucene) to behave like library metasearch systems or commercial systems (like A9).

4.1 Experimental Design Sketch for Known Item search

The sketch is as follows:

1. Give each user a random resource from the DL.
2. Have them look for the resource through each search interface.
3. When they find it, they can click on a "found it" button to indicate success.
4. Throughout this process, record the number of queries it took (and what they were), as well as the position of the result. Record how long it took, etc.
5. Ask their impressions after resource they find, so they can talk about reactions to the different search engines.

For the this kind of evaluation, we are concerned that searches will be "too easy" if users are able to simply enter in parts of the actual metadata. Normally, even known-item searches are based on imperfect recall of the record's information.

4.2 Experimental Design Sketch for Open-Discovery Search

1. Give participants a topic they have to learn about (students would be good for this).

2. Give them a DL with content that covers the topic. Content should be of varying types and quality. Need learning artifacts that will work for short-term sessions.

3. Have people search for resources to learn about the topic, given one of the search interfaces (could be arbitrary).

4. Quiz/test them on the topic when they are ready. Record how long it took for them to start the quiz, and how well they do on it (the topic should be nontrivial enough that it won't be too easy).

5. Compare the results between search systems. Can also look at the learning effects of each search system, through time.

For this evaluation, it would be a challenge to come up with good topics that the participants can learn about within the span of the experiment, yet still show a good distribution of results on the evaluation "quizzes."

5 QM-Search Development

In this chapter we discuss the development of the QM-search prototype system, a process being undertaken as a modification of the Lucene system. This seciton is split into two parts. The first summarizes the current state of the design of this system. The second part discusses the progress to date toward implementing that design.

5.1 Architecture

5.1.1 Introduction

This section is intended to provide a description of the technical design of the QM-search system, now being implemented.

5.1.2 A Typical Usage Scenario

The computer science department of Sprayberry High school has a presentation on Intel's hyper-threading technology by Prof. Brown. Even though Prof. Brown has extensive knowledge of the technology, he needs more details like history information, background and exact dates to prepare his presentation.

He opens Emory's digital library search engine. This search engine supports different profiles for different needs. He loads the scholar's profile and searches for hyper-threading. Since this is a profile for the scholars, it configures the search engine to filter the results based on "rating," rating to be always greater than 50%, the results be ranked on "expertise level" and displayed in decreasing level of complexity. He gets the information he needs in the first few results of his search. He can also divide the search result into two domains, image or text, further making his task fast and efficient.

Jenny, a high school student is planning to attend this presentation. Before going to the talk she wants to have a fair idea of the topic but does not want to go into the details. So she opens Emory's digital library search engine and loads student profile to search for hyper-threading. Since this profile caters to high school students, it is configured to rank the result based again on expertise level (this time in increasing order of complexity) and the results filtered based

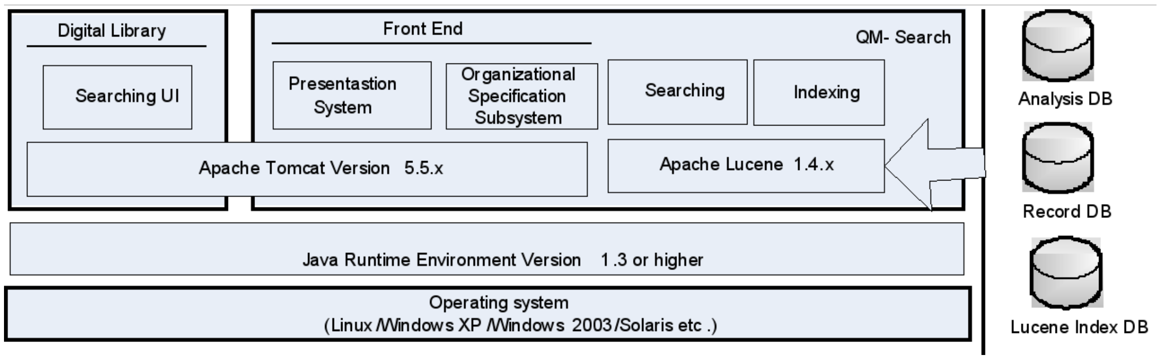

Figure 5.1: Overview of QM-search, with connected system dependencies.

on popularity (greater than 80%) and clarity (greater than 75%). The results from the search engine fulfill Jenny's needs.

Even though both users use the same search engine, it can cater to different needs.

5.1.3 Software Dependencies

The system software dependencies for QM-search System are as follows:

1. Any Operating system which can support Java Runtime Version 1.3 (recommended Operating System: Debian Linux Version 3.1)

2. Java Runtime Environment for the selected OS (required Version: 1.3 or higher). See `http://java.sun.com`.

3. Apache Tomcat (recommended Version: 5.5.9 or higher). See `http://jakarta.apache.org/tomcat/index.html`.

4. Apache Lucene (required Version: 1.4 or higher). See `http://lucene.apache.org`.

How all of these dependencies are situated with the QM-search system is shown in Figure 5.1.

5.1.4 Development Environment

The indexing and searching service is being developed on a Debian Linux-based system with Apache Tomcat application server.

Apache Lucene is being used as the base library for the search engine. Lucene is a Java-based open source search engine library, a part of the Apache project.

5.1.5 Architecture

The Search System can be broadly divided into 4 modules:

1. **Indexing**

   The indexing module indexes the records by converting the records to Lucene documents and making them searchable on content and indicators. Indexing already exists in Lucene; we are simply augmenting this portion.

2. **Organization Specification Subsystem**

   This module allows users/DL deployer to specify filtering and dimensional specifications. These specifications would be used to filter search results and organize ranked search results in various specified logical "dimensions."

3. **Searching**

   The searching module provides searching capability on the indexed records and outputs search results based on scoring criteria. This module also already exists and is the core functionality of Lucene; are simply generalizing it here.

4. **Organization Output Subsystem**

   This module organizes the result and returns it in an XML format which can then be rendered in arbitrary ways by the digital library environment.

Details on the modules are in the following sections.

5.1.6 Indexing

The indexing system can be divided into 2 sub-systems:

1. Analysis sub-system
2. Indexing sub-system

The sub-systems are described as follows:

Analysis Subsystem

This system will perform - batch calculations and/or data mining from record attributes and auxiliary data prior to indexing and output indicators.

Input :	Record attributes from different sources
	Auxiliary data
Output :	Indicators

The system will allow the DL deployer to select the records they want to process through the analysis system. The indicators generated by the Analysis system is stored into the system will be used for final indexing. An overview of the analysis system and its connection to indexing and configuration is given in Figure 5.2. A diagram of the system itself is shown in Figure 5.3.

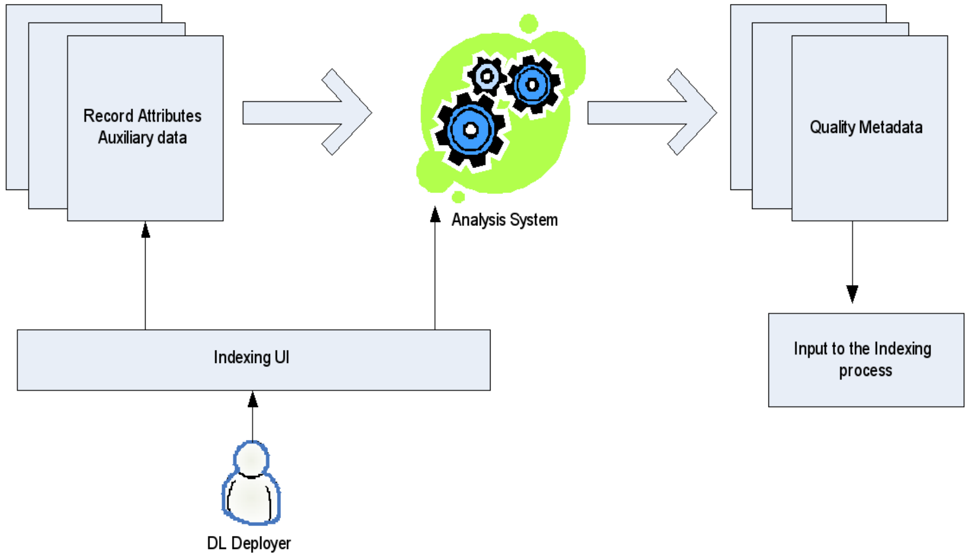

Figure 5.2: Overview of the indexing analysis system, with its relation to the indexing and DL deployer configuration process
.

Indexing Subsystem

The indexing subsystem converts the attributes and indicators from the analysis system to data in Lucene-compatible document objects. This allows the search system to index and the information and use it in the scoring process, without modification of the document object class.

Input:	Content Metadata
	Indicators (Analysis subsystem output)
Output:	Indexed records

Flow for the Indexing Process

The flow for the indexing process is as follows:

1. Index the digital library metadata normally in Lucene, creating a first-cut index.

2. The analysis subsystem reads in the first-cut index, and uses analysis algorithms (plug-ins) to determine new quality indicators.

3. Augment the first-cut index with quality indicators, creating a final-cut index.

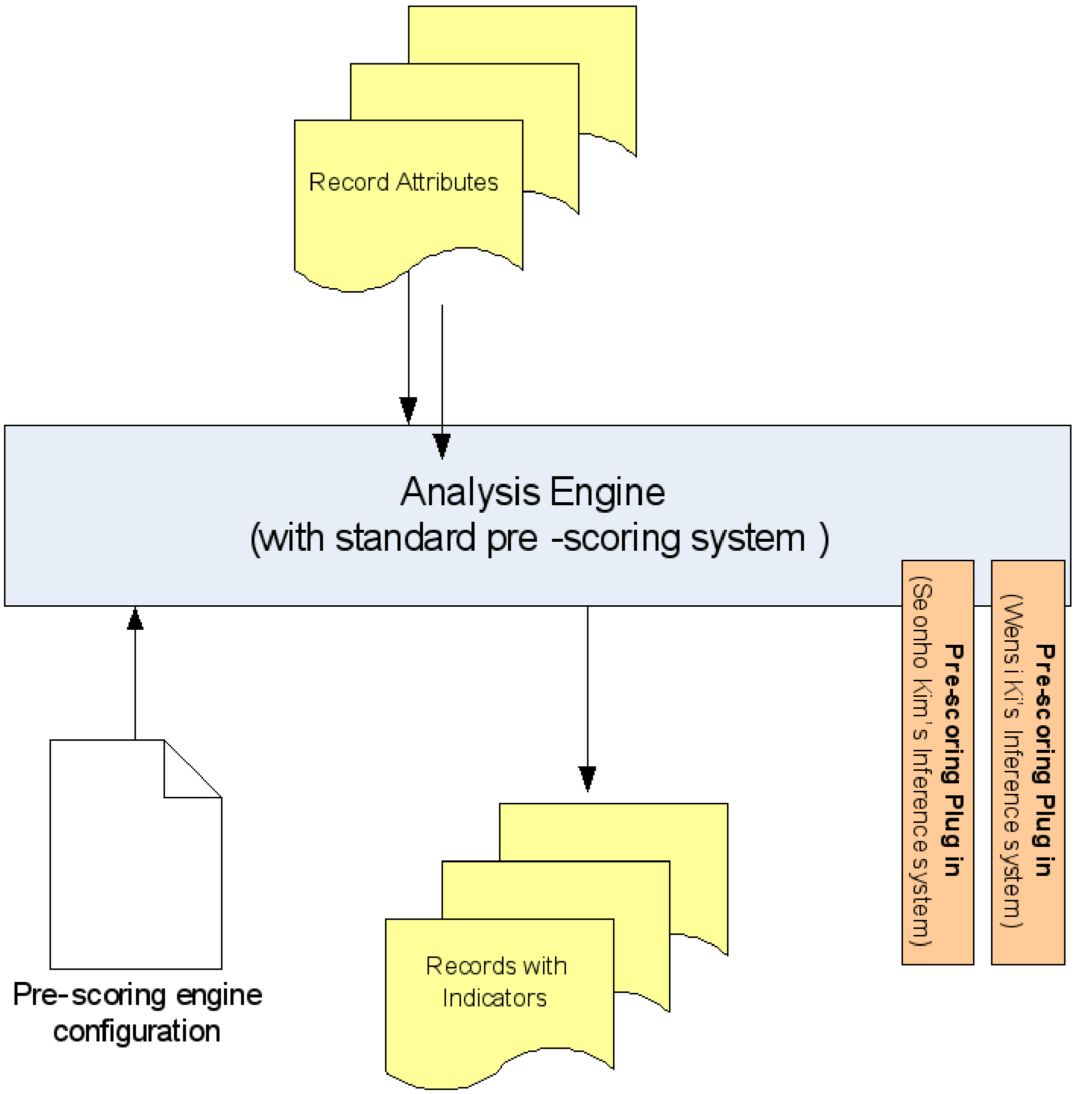

Figure 5.3: A schematic of the analysis/pre-scoring system.

Initial Static Model for Indexing

1. **Class QMSearchAnalysisEngine**

   Class Definition:

QMAnalysisEngine
-DigitalLibraryRecordSet : DigitalLibraryRecord
+AddRecords() +AddRecord() +RemoveRecords() +SetRecordAt() +ProcessRecord() +ReadPreScoringPlan() +ListRegisteredPlugins()

   Description:

   Class QMAnalysisEngine is the primary class responsible for execution analysis engine. It is designed to support a configurable and extensible system to determine indicators for the records to be indexed.

   The analysis engine is expected to be implemented as a background process which can be invoked by the indexing user interface to analyze and index records. The engine will use a user defined plan while creating indicators for the record.

2. **Interface IQMAnalysisPlugIn**

   Class Definition:

IQMAnalysisPlugin
+ListKeys() +AddKey() +RemoveKey() +UpdateRecordAttribute() +SetRecord() +GetRecord() +GetPluginName() +GetPluginDescription()

   Description:

   Interface IQMAnalysisPlugIn is the primary interface class to be implemented by all analysis plug-in classes to allow them to be used by the analysis engine.

3. **(Utility Class) AttributeNameValuePair**

   Class Definition:

#AttributeNameValuePair : string(idl) #AttributeValue : any(idl)
+GetAttributeName() : string(idl) +SetAttributeName(in keyName : any(idl)) +GetAttributeValue() : any(idl) +SetAttributeValue(in keyValue : any(idl))

Description:

This class is storage class for name-value pair for record attributes.

4. **(Utility Class) DigitialLibraryRecord**

   Class Definition:

-AttributeList : sequence(idl)
+AddAttribute(in attributeKeyvalue : AttributeNameValuePair) +RemoveAttribute() +SetAttributeAt() +SetAttributeValue()

   Description:

   This class is storage class for digital library record which is typically constitutes of list of record attributes.

5.1.7 Organizational Specification Subsystem

This subsystem allows the DL deployer to specify various filtering and dimension criteria. This information serves as an input to the search system to produce the final ranked search results grouped according to the specified dimensions.

Input:	Filtering criteria and dimensions in XML.
Output:	Updated user query
	Dimensional specifications

The DL deployer can specify different specifications (filtering criteria and dimension) for different profiles.

The profiles in the scenario described at the beginning of this section could be rendered as organization specifications as shown in Figures 5.4 and 5.5. Note the different dimensions of organization declared, and how how results are filtered and grouped differently.

Design for the Organization Specification Subsystem

This subsystem reads filtering and dimension information provided by the DL deployer. The filtering specification is appended the user query to generate a combined query which is used for generating search results. Dimensional specifications are input to the search module to rank results along the various dimensions specified by the deployer.

```
<organization>
    <filter>
         <fil_and>
               <clause metadata="popularity" function="range" value="[0.8 TO 1]"/>
                <clause metadata="clarity" function="range" value="[0.75 TO 1]"/>
          </fil_and>
     </filter>
    <dimension>
           <key>
                <metadata>expertise_level</metadata>
            </key>
        <sort order="ascending"/>
        <binning type="natural"/>
    </dimension>
</organization>
```

Figure 5.4: A sample organization specification, based on the “student” profile.

```
<organization>
    <filter>
         <clause metadata="rating" function="range" value="[0.5 TO 1]"/>
    </filter>
    <dimension>
           <key>
                <metadata>expertise_level</metadata>
            </key>
        <sort order="descending"/>
        <binning type="fixed">15</binning>
    </dimension>
</organization>
```

Figure 5.5: A sample organization specification, based on the “scholar” profile.

Initial Static Model for Organization Specification Subsystem

1. **Class QMGetFilterSpecs**

   Class Definition:

QMGetFilterSpecs
+generateFinalQuery(in userQuery : string(idl), in XmlSpecification : string(idl)) : string(idl)

   Description:

   > Class QMGetFilterSpecs is the class responsible for reading filter section of organization specifications provided by the deployer and converting it to a query that can be appended to the user query. This combined query will be used by the search module to calculate get results.

2. **Class QMGetDimensionSpecs**

   -TBD

5.1.8 Searching

This system provides a search interface to the indexed records. This module should be called n number of times if n dimensions are provided by the DL deployer, each time with a different dimension.

Input:	Search criteria (User Query + Deployer specification)
Output:	Sets of collection of records which met the search criteria and ranked based on the dimensions provided in organization specifications.

Design for searching system

The search system extends the Scorer and Similarity classes to customize the searching interface provided by Lucene. Dimensions as specified by DL deployer will be provided as input to rank the search results.

5.1.9 Organization Output Subsystem

This subsystem groups the result of the user query provided by Lucene's output based on the output specifications selected by the user and coverts it into XML format for display (for example using XSLT transformation).

Input:	Instance of class Hits (Lucene class in which it return the result of the search)
Output:	XML Document

Initial Static Model for the Organization Output System

1. **Class QMXMLConverter**

   Class Definition:

QMXMLConverter
+OrganizeDataAsXML() : XMLDocument

   Description:

   Class QMXMLConverter is the primary class responsible for converting Lucene's output to XML document to be displayed to the user of search engine.

For example, the user can ask for the content query results to be organized in 4 groups using 2 dimensions and selects the DL deployer profile of the scholar. One dimension being the "source" which can be either "publications" or "blogs" and second dimension being the year of publishing grouped as "before year 2000" or "2000 and after."

An example of organized output, which is transmitted as structured XML, is given in Figure 5.6.

5.2 Progress Report

The progress in developing QM-search to date consists of the following points:

1. Evaluated feasibility of Apache Lucene to be configured as a search engine for the project.

   - Downloaded and Configured Apache Lucene version 1.4 on Debian/Linux platform and Java Runtime Environment (JRE) 1.4.
   - Developed a prototype search engine using Java Development Kit (JDK) 1.4 and Lucene libraries.
   - Tested indexing and search features against sample set of records

2. Prepared system architecture document for QM search engine. The architecture is based on our earlier "Plan for modifying Lucene to Implement QM-search System."

   - The central features of this architecture include:
     - Support for various plug-ins for Analysis subsystem.
     - Support for records with configurable properties.
     - Support for different deployers profiles for different users and needs.
     - Allows Dynamic selection of deployer profile during run time.
     - Ability to save search results as XML files

```xml
<results>
        <item id = "Id1">
                <dim ord = "1" name = "..." score=".99"/>
                <dim ord = "2" name = "..." score=".34"/>
                <dim ord = "3" name = "..." score=".55"/>
        <metadata>
                        <author>Vikram Seth</author>
                        <category>fiction</category>
                        ...
                        ...
        </metadata>
        </item>

        <item id = "Id2">
                <dim ord = "1" name = "..." score=".10"/>
                <dim ord = "2" name = "..." score=".34"/>
                <dim ord = "3" name = "..." score=".59"/>
        <metadata>
                        <author>Salman Rushdie</author>
                        <category>drama</category>
                        ...
                        ...
        </metadata>
    </item>
    ...
    ...
    ...
</results>
```

Figure 5.6: An example of organized output of QM search, which is given as an XML document.

- Designed validation schema for the deployer profile to allow users to validate their configuration.

3. Implementation Progress

   - Implemented Class loading feature to load plug-ins for Analysis subsystem.
   - Created module to interpret and convert search output to XML format.
   - Implemented selection of deployer specification by the users from a pool of available profiles at runtime
   - Developed parser to process deployer's filter specifications.
   - Coded to processed deployer's filter specifications.
   - Implemented schema to validate deployer's filter specifications.

Next steps:

- Design and implement a web front end for the search interface
- Design and implement harvester to import data into the indexing system.
- Implement dimensional specs provided by the deployer.
- Design and deploy XML style sheets to display search results.

6 Progress Report from Virginia Tech

(By Vikram Raj)

The work done at Virginia Tech has been mostly oriented towards theoretical preparation for the experimental setup to be done in Spring 2006. Also, I have been interacting with people in the Digital Libraries Research Laboratory (DLRL) who have been engaging students in user studies.

I have been learning on 5S modeling techniques that can be used to form a theoretical model of a digital library, and associated DLGen, a system that would generate a barebones digital library from a given meta-model.

Also, I have been reading on different logging schemes that have been developed by other members in the DLRL for other digital libraries. These logging schemes might make usability testing easier, as the behavior of the user need not be constantly monitored, and the videos observed minutely. In this scheme, the videos would serve more as a due-diligence record-keeping and professionalism provision.

In parallel, we have been making the documents required for approval by the Institutional Review Board (IRB) of Virginia Tech. In this regard, I have undertaken courses and certification to conduct user studies involving human studies. The documents still need some screen dumps to be put in, before being submitted. Rohit Chopra provided a great deal of help by lending the documents that he had used for submission to the IRB at Emory University.

By January, we plan to let word out to students and professors for recruitment, and then get started with experiments by middle or end of February.

Rohit Chopra, Urvashi Gadi, and I have been in frequent contact about developments in the project. I made a trip to Emory University in November 2005, and got to meet the personnel involved over there. I also got to see a video recording of a session that was aimed at getting qualitative data. I got a feel of what was being expected to be extracted at the end of qualitative studies. I realized that qualitative and quantitative studies could provide feedback to one another. I also came to know that novice users would also explore a tool in detail, if they liked it.

On another front, I have been involved in setting up some infrastructure that might be used for providing data for IMLS. Work on this should be complete by middle of December.

Besides than these activities, we have been involved in critiquing and providing feedback on interfaces that were designed by Emory for use in the focus group studies.

7 Data Sources

In order to model and experiment with quality metrics in metasearch, as is the goal of this project, the underlying data from which quality indicators are derived is needed. While some of this data is latent in typically-available metadata fields (such as publication and author-related indicators), deeper quality indicators (such as citation-, selection-, or activation-based indicators) typically require gathering information from secondary sources.

Thus, a continuous thread of activity in this project has been determining which such sources are available to us, for the digital library records we have, so that we might derive quality indicators from them for testing, or at least model the data and learn what is possible in typical digital library scenarios.

To pursue this objective, we have met with a number of librarians here at Emory who are experts in various aspects of our digital library holdings and digital aspects of our library infrastructure. These people have primarily been:

- Selden Deemer - Library Systems Administrator (esp. regarding our EUCLID catalog, SFX, and EZProxy.)
- Bernardo Gomez - Library Systems Administrator (same as above.)
- Jason White - Software Developer (esp. regarding ReservesDirect.)
- Lisa Macklin - Electronic Resources Team Leader (esp. regarding electronic holdings and contracts.)
- Kyle Fenton - Head Systems Administrator for Digital Projects (esp. regarding SouthernSpaces and the AmericanSouth digital libraries.)

Many of these people were consulted in the grant proposal phase as well. From discussions with these experts, we have identified quality indicator data sources which are broken down in Table 7.

We also plan to explore other possibilities, and consult with electronic journals and databases providers. Hopefully, these vendors will be willing to work out arrangements whereby we can use auxiliary data of high likihood to contain quality information at no cost, at least for research purposes. Such potential data, which is commercially and technically inaccessible to us at the moment, typically includes citation linkage information, as well as review/rating or recommendation data. It is likely Lisa Macklin will be involved in any contractual negotiations with vendors necessary to get access to such data.

Table 7.1: Likely sources within the Emory library environment for quality indicator information.

Data source	Quality information type	Contact(s)	Detailed Description
EUCLID circulation logs	activation	Bernardo Gomez	EUCLID is the library's main circulation system. Anonymized versions of logs are available which indicate which item was checked out or renewed, what type the item is, and by what user class.
ReservesDirect (live) database	selection/ inclusion	Jason White	ReservesDirect is Emory's e-Reserves system. It allows faculty to place items on reserves for courses. This system is rich with information about the desireability of particular books or articles. However, the metadata entered by faculty is uneven and deduping will be a challenge.
SFX (journal) usage reports	activation	Selden Deemer	SFX is our unified electronic holdings gateway. All access to electronic holdings is recorded and electronic reports are accessible, from which the popularity of particular targets (publishers) or periodicals is available. From this information, publication quality indicators could be derived. However, extracting this information comprehensively will be a challenge.
EZProxy usage logs	activation	Bernardo Gomez	EZProxy is the first step in access to all electronic holdings, and is a consortium system with all Georgia institutions of higher education. The HTTP-style logs generated by EZProxy can be mined for activation information of individual articles. Producing a parser for these logs looks to be a challenge.

8 Work Plan

The high-level tasks we must undertake to completion of this project are given below, by quarter.

8.0.1 Q3 2005 (passed)

- Finalize selection of search engine upon which to build QM-search.
- Continue research into scoring model techniques.
- Continue to develop the scoring and presentation models.
- Detail how to map these models into the specific search system.
- Begin implementation of search system.

8.0.2 Q4 2005 (current)

- Complete search system implementation (initial version).
- Detail focus groups (gather exemplary system, define "seed" questions).
- Design experimental evaluation of QM-search system.
- Execute focus groups.

8.0.3 Q1 2006

- Execute focus groups (multiple rounds).
- Execute quantitative experiments.
- Refine search system.

8.0.4 Q2 2006

- Analyze and report on results of focus groups; work into theoretical models.
- Analysis of quantitative experiments.
- Refine search system.

8.0.5 Q3 2006

- Conclude analysis.
- Reporting (peer and project).

Bibliography

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998. URL `http://citeseer.ist.psu.edu/brin98anatomy.html`.