Evaluation and Evolution of a Browse and Search Interface: Relation Browser++

Junliang Zhang
Interaction Design Laboratory
School of Information and Library Science
University of North Carolina, Chapel Hill
919-962-8274
junliang@email.unc.edu

Gary Marchionini
Interaction Design Laboratory
School of Information and Library Science
University of North Carolina, Chapel Hill
919-966-3611
march@ils.unc.edu

ABSTRACT

We present in this paper the design and an evaluation of a novel interface called the Relation Browser++ (RB++) for searching and browsing large information collections. RB++ provides visualized category overviews of an information space and allows dynamic filtering and exploration of the result set by tightly coupling the browsing and searching functions. A user study was conducted to compare the effectiveness, efficiency and user satisfaction of completing various types of searching and browsing using the RB++ interface and a traditional form-fillin interface for a video library. An exploration set of tasks was also included to examine the effectiveness of and user satisfaction with the RB++ when applied to a large federal statistics website. The comparison study strongly supported that RB++ was more effective, efficient, and satisfying for completing data exploration tasks. Based on the results, efforts to automatically populate the underlying database using machine learning techniques are underway. Preliminary implementations for two large-scale federal statistical websites have been installed on government servers for internal evaluation.

Categories and Subject Descriptors

:**H.5.2 [Information Interfaces and presentation (e.g. HCI)]: User Interface -** ***interaction style, graphical user interfaces (GUI);*****H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval -** ***query formulation, search process, selection process;*****H.3.7 [Information Storage and Retrieval]: Digital Libraries – User issues, Systems issues**

General Terms:

Design, Human Factors, Experimentation, Performance

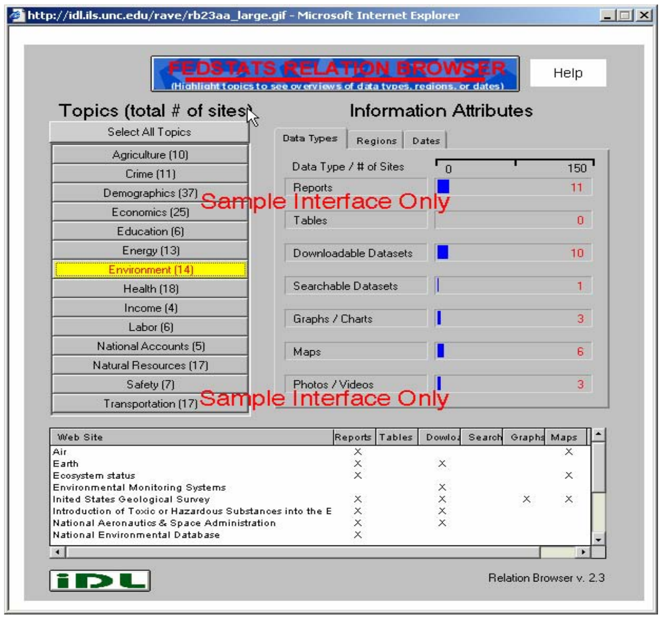
Keywords:

Interface design, search, browse, category overview, visualization, dynamic query, interactive system

1. INTRODUCTION

The size and breadth of large government websites and digital libraries makes it difficult for people to quickly grasp what content is and is not available. Dynamic overviews and previews of collections can help people decide if it is worthwhile to look further [8]. As they do look further, it is helpful for their searching and browsing to quickly see partitions of the collection and how many items are available in different partitions. We believe that government website users will be well-served by highly interactive user interfaces that support alternative views of the collections, partitions, and results sets. This paper describes a user interface that aims to provide agile control for browse and search, reports results from a user study comparing this interface to a typical WWW search interface, and describes the ongoing evolution of the interface, including efforts to automate discovery of topical categories and assignment of webpages to those categories.

Faceted category structure is one way to help people understand the composition of an information collection. A faceted approach provides different ways to slice and dice the information space, which allows people to look at the information space from different perspectives. Allowing people to explore the relationships among different facets may further deepen their understanding and support new insights. The relation browser (RB) is an interface which provides an overview of the collection by displaying different categories and enables people to explore the relationships among these categories [13]. The different facet values also serve as selectable objects that may be employed as query widgets for a search so that the entire space can quickly be partitioned with simple mouse moves and with consequent immediate display of the resulting partition in the results panel. Figure 1 shows the mock-up interface of an early version of the relation browser in the domain of U.S. federal statistics websites. The web pages in the site were sliced into four different facets: by topic, data type, region, and date. The numbers beside the bars indicate the number of websites associated with the attributes. By mousing over any of the topics, the distribution of the specific topics in other facets are visualized as graphic bars. The underlying data for this instance of the interface was manually extracted from a small set of 200 webpages contained in more than 70 federal statistical agency websites.



Web Site	Reports	Tables	Dowloa	Search	Graphs	Maps
Air	X					X
Earth	X		X			
Ecosystem status	X					X
Environmental Monitoring Systems			X			
Inited States Geological Survey	X		X		X	X
Introduction of Toxic or Hazardous Substances into the E	X		X			
National Aeronautics & Space Administration	X		X			
National Environmental Database	X					

Figure 1. Relation Browser (RB)

This early version of the RB has been redesigned based on user studies and experience applying the interface to more than a dozen different database instances [13]. The new version is called RB++, which improves the RB significantly in several ways (see Figure 2) [23,24]. First, RB++ displays multiple facets (categories) visually and on the same screen rather than only two facets with tab options to others. The multiple facets provide an overview of the information space. The facet values are visually represented by graphic bars with different lengths, which indicate the number of items associated with them. Second, RB++ allows more flexibility to explore relationships. One of the features of RB++ is that you can restrict the information items (partition the information space) by mousing over any bars and other bars are proportionally highlighted to show the conditional distribution across all the facets. Note that the previous RB was limited to visualizing pairwise relationships with one main facet. Third, the RB++ added a dynamic filtering function for the result set (see Figure 3). Once the search results are displayed in the table, further filtering can be done by typing in keywords (string patterns) in the boxes located immediately above the result fields. The filtering is dynamic, which means that with each character typed in or removed from the boxes, RB++ matches the string patterns in the boxes with the corresponding field of the results. Only the matched results are then displayed immediately in the results panel and the matched string in the results is highlighted. This dynamic feature gives users instant and constant feedback about the filtered results and how many items they will get with different keywords, which allows users to try out different filtering keywords very easily and efficiently. Fourth, the RB++ provides an overview of the results set and tightly couples the overview and results set panels. The overview panel is dynamically updated to give users a contextualized overview of the updated result set. These new features give users more power to understand and explore the information collection and give them a flexible and rapid way to find the information they want. A linguistic model of BNF grammar to model the user interaction with the interface is provided in section 2.3 to help reveal the dynamic nature of the RB++.

In the paper, we argue that the RB++ interface will bring users added values beyond simple searching and browsing by in fact combining these search strategies seamlessly. In the next section, the methodology of a user study is described. The results of the user study are then presented and discussed. Limitations of the interface and current efforts to deal with data classification are then described.

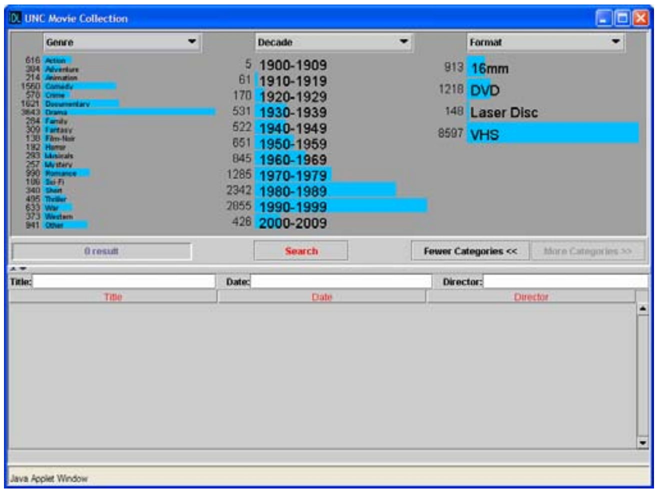
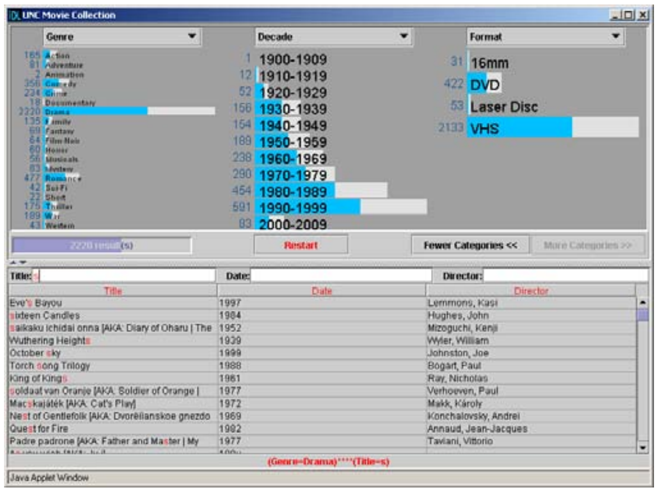


Figure 2. Initial display of RB++ with visualized category overview on the top



Title	Date	Director
Eve's Bayou	1997	Lemmons, Kasi
sixteen Candles	1984	Hughes, John
saikaku ichidai onna [AKA: Diary of Oharu | The	1952	Mizoguchi, Kenji
Wuthering Heights	1939	Wyler, William
October sky	1999	Johnston, Joe
Torch song Trilogy	1988	Bogart, Paul
King of Kings	1961	Ray, Nicholas
soldaat van Oranje [AKA: Soldier of Orange |	1977	Verhoeven, Paul
Macskajáték [AKA: Cat's Play]	1972	Makk, Károly
Nest of Gentlefolk [AKA: Dvorëlianskoe gnezdo	1969	Konchalovsky, Andrei
Quest for Fire	1982	Annaud, Jean-Jacques
Padre padrone [AKA: Father and Master | My	1977	Taviani, Vittorio

(Genre=Drama)****(Title=s)

Figure 3. RB++ with dynamic filtering of the results (note the changes in the overview and updated results)

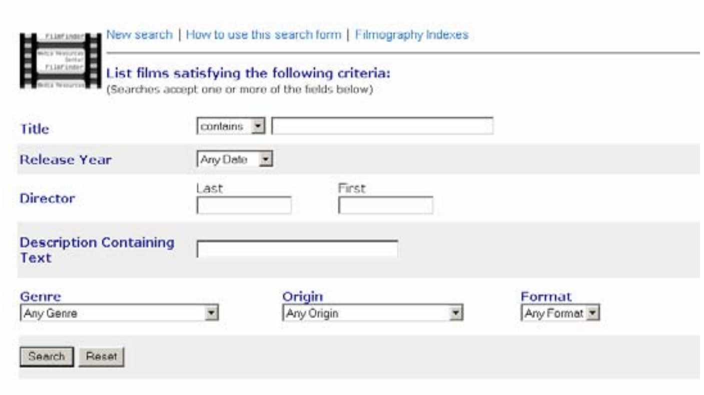
2. METHODOLOGY

The purpose of the user study was two-fold: first, we wanted to compare the effectiveness, efficiency and user satisfaction associated with completing certain tasks using RB++ against that obtained by the traditional form-fillin search interface (baseline interface). Second, we wanted to explore if the RB++ interface would lead to new interaction patterns with the interface and if so, to determine what these new interaction patterns might be.

Seventeen undergraduate and graduate students were recruited from the UNC-Chapel Hill campus for this study. They came from various schools and departments. There were 10 females and 7 males with an age range from 19 to 44, (15 were in their 20s, and all were familiar with www browsers. The participants were given $15 for their participation. The data from the first two participants was used as a pilot test; based on which the experimental protocol and instruments were revised. The data from the other 15 participants was used for the data analysis. The study proceeded in two phases, a within subjects comparison across two different interfaces for the same database, and an exploratory investigation with a single government statistical website instance of RB++.

2.1 Phase One: RB++ to Baseline Comparison for a Film Database

The first phase was a comparison study in which participants used both the RB++ interface and the baseline interface. The order of using these interfaces was counter balanced. The domain of the information items in both interfaces was the video collection in the UNC-CH library (http://www.lib.unc.edu/house/mrc/index.html?page=filmography) that contains about 10000 films. The library online video search interface (Filmfinder) was used as our baseline interface. FilmFinder is a fairly typical www form-fillin search interface (see Figure 4 and Figure 5), where users can specify queries within fields such as title, release year, director, description, genre, origin, and format.



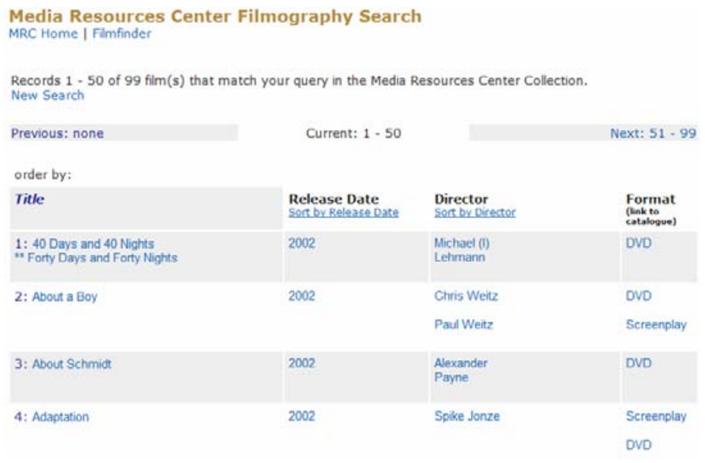
New search | How to use this search form | Filmography Indexes

List films satisfying the following criteria:
(Searches accept one or more of the fields below)

Title	contains
Release Year	Any Date
Director	Last / First
Description Containing Text	
Genre	Any Genre
Origin	Any Origin
Format	Any Format

Search | Reset

Figure 4. Filmfinder with Form-fillin Interface



Media Resources Center Filmography Search
MRC Home | Filmfinder

Records 1 - 50 of 99 film(s) that match your query in the Media Resources Center Collection.
New Search

Previous: none Current: 1 - 50 Next: 51 - 99

order by:

Title	Release Date Sort by Release Date	Director Sort by Director	Format (link to catalogue)
1: 40 Days and 40 Nights ** Forty Days and Forty Nights	2002	Michael (I) Lehmann	DVD
2: About a Boy	2002	Chris Weitz Paul Weitz	DVD Screenplay
3: About Schmidt	2002	Alexander Payne	DVD
4: Adaptation	2002	Spike Jonze	Screenplay DVD

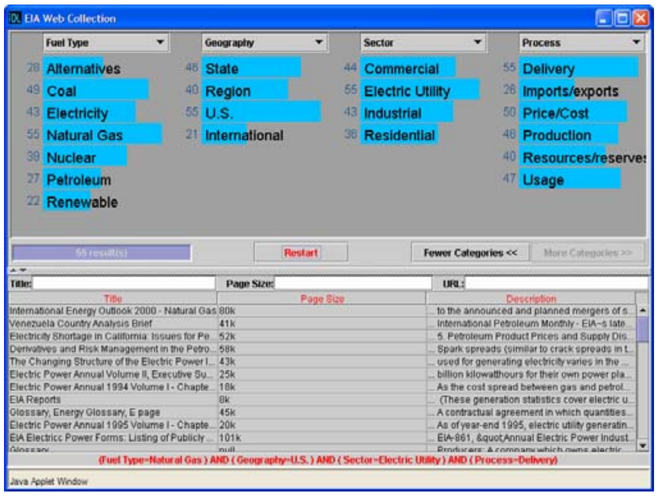
Figure 5. Results Page of Filmfinder

All participants were run individually in sessions ranging from 60-90 minutes and all sessions were video taped. The protocol for the first phase was as follows: First, a demographic pre-test questionnaire was completed. Second, the participant was trained for the first interface assigned in their condition. The training consisted of: an introduction to the features of the interface, a demo of each type of task with the interface, and participant practice using the interface until s/he was comfortable with it. Third, the participant used the interface to complete 10 search tasks. Tasks were assigned to participants one by one by handing them pieces of paper for each task. A timer was used to count time used to complete each task except for task 10 (see description of task 10 below). After each task, a short satisfaction questionnaire was completed by the participant. Fourth, a usability questionnaire was filled out after the participant finished using the first interface. Next, the participant was trained for the second interface and the same procedures were used to complete 10 more search tasks. Finally, an open-ended questionnaire about perceived differences and preferences for the two interfaces was completed.

The tasks were classified into three different types: 1. Simple look up task. Tasks 1 to 3 in each task set were of this type. For example, "Check if the movie titled "The Matrix" is in the library movie collection." 2. Data exploration and analysis tasks. Tasks 4 to 9 in each task set were of this type. This kind of task requires users to understand and make sense of the information collection, which could be a starting point for them to further their searching or browsing. Two examples of this type are: "In which decade did "Steven Spielberg" direct the most movies?"; and "How many movie titles does the library hold that were released in the year 2000?" 3. Task 10 was a free exploration task, which asked participants to find five favorite videos without any time constraints. The tasks assigned for the two interfaces were different but comparable. For example, the comparable tasks for two interfaces simply substituted different video titles or directors.

2.2 Phase Two: Explore RB++ for EIA Website

The second phase was an exploratory study of the RB++ applied to roughly 10,000 pages in the Energy Information Administration (EIA) website. Based on intensive manual inspection of the EIA website, four facets were identified with associated facet values: fuel type (with the facet values: alternatives, coal, electricity, natural gas, nuclear, petroleum, and renewable); geography (state level, regional level, national level, and international level); sector (commercial, electric utility, industrial, and residential); and process (delivery, import/export, price/cost, production, resources/reserves, and usage). All the facets were displayed on the overview panel (see Figure 6). The results panel displayed the title, page size, and description of the web pages.



Title	Page Size	Description
International Energy Outlook 2000 - Natural Gas	80k	...to the announced and planned mergers of s...
Venezuela Country Analysis Brief	41k	...International Petroleum Monthly - EIA~s late...
Electricity Shortage in California: Issues for Pe...	52k	...5. Petroleum Product Prices and Supply Dis...
Derivatives and Risk Management in the Petro...	58k	...Spark spreads (similar to crack spreads in t...
The Changing Structure of the Electric Power I...	43k	...used for generating electricity varies in the ...
Electric Power Annual Volume II, Executive Su...	25k	...billion kilowatthours for their own power pla...
Electric Power Annual 1994 Volume I - Chapte...	18k	...As the cost spread between gas and petrol...
EIA Reports	8k	... (These generation statistics cover electric u...
Glossary, Energy Glossary, E page	45k	...A contractual agreement in which quantities...
Electric Power Annual 1995 Volume I - Chapte...	20k	...As of year-end 1995, electric utility generatin...
EIA Electricc Power Forms: Listing of Publicly ...	101k	...EIA-861, &quot;Annual Electric Power Indust...
Glossary	null	Producers: A company which owns electric

(Fuel Type=Natural Gas) AND (Geography=U.S.) AND (Sector=Electric Utility) AND (Process=Delivery)

Figure 6. RB++ interface applied to EIA website

The protocol of the second phase was as follows: First, the RB++ EIA application was introduced to the participant. Second, the participant practiced using the interface until s/he was comfortable with it. Third, the participant used the interface to complete four tasks[1]. The process was recorded and a short satisfaction questionnaire was filled out after finishing each task. Fourth, an open-ended questionnaire was completed after finishing all the tasks. Lastly, the participant was briefly interviewed.

Data collected included both quantitative and qualitative data from the two phases of the study. Data collected for the first phase included performance data (time spent finishing tasks), error rates of tasks, ratings on the satisfaction questionnaire after finishing each task, ratings on the usability questionnaire after finishing each interface, and comments on the open questionnaire about perceived differences and preferences for the two interfaces. Data collected for the second phase included ratings on the satisfaction questionnaire after finishing each task, comments on the post-session questionnaire and the verbal comments made in the interview.

[1]Tasks for the second phase study:
1. I want to learn the current status of Chinese nuclear energy.
2. Find the most recent weekly data on petroleum prices in the USA.
3. Find the statistical data on coal production across different states in the year 2001.
4. What kinds of information can I and can I not find from the website?

2.3 Modeling User Interaction

To help us form hypotheses and analyze and make sense of the experimental data, we employed a linguistic model, called BNF grammar, to model the user's interaction with the interface. BNF grammar was originally used by Reisner to describe the dialog grammar of an interactive graphics system [15], where the user's interaction with a system was seen as an action language and BNF grammar was used to formally describe the language. The BNF grammar consists of a set of rules which define higher level user behaviors in terms of lower level ones. Each rule can be composed of terminals, non-terminals, and a set of symbols. Terminals usually represent the lowest level of user behavior, such as pressing a key or clicking a mouse button and can not be further defined. Non-terminals represent a high level abstraction and can be defined in terms of other non-terminals and terminals. Terminals are written with upper case letters and non-terminals are written with lower case letters. The "::=" symbol is read as " is defined as". The "+", "|" and "-" symbols are used at the right hand side of rules to connect, respectively, sequence of user behavior, set of options, and concurrent user behaviors. With the BNF grammar, we can describe the user's interaction with the RB++ as follows:

A1 information seeking ::= explore collection(**A3**) | (formulate query(**A2**) + CLICK SEARCH BUTTON + navigate results(**A5**))

A2 formulate query ::= (explore collection(**A3**) + form query(**A4**)) | form query(**A4**)

A3 explore collection ::= (CLICK VISUAL BAR-OBSERVE VISUAL BAR + explore collection(**A3**)) | (MOUSE OVER VISUAL BAR-OBSERVE VISUAL BAR + explore collection(**A3**))

A4 form query ::= (CLICK VISUAL BAR + form query(**A4**)) | (TYPE IN KEYWORD + form query(**A4**))

A5 navigate results ::= (browse results(**A6**) + navigate results(**A5**)) | (CLICK RESTART BUTTON + information seeking(**A1**))

A6 browse results ::= (show results(**A7**)-OBSERVE RESULTS + browse results(**A6**)) | (CLICK RESULT ITEM + browse results(**A6**)) | (CLICK SORTING BUTTON + browse results(**A6**))| (explore results(**A8**) + browse results(**A6**))

A7 show results ::= CLICK SIDEBAR

A8 explore results ::= (observe system state(**A9**) + explore results(**A8**)) | (filter results(**A10**) + explore results(**A8**))

A9 observe system state ::= (OBSERVE VISUAL BAR + observe system state(**A9**)) | (OBSERVE NUMBER + observe system state (**A9**))

A10 filter results ::= CLICK VISUAL BAR | MOUSE OVER VISUAL BAR | TYPE IN KEYWORD

The interaction with baseline interface can be described as:

B1 information seeking ::= formulate query(**B2**) + CLICK SEARCH BUTTON + navigate results(**B4**)

B2 formulate query ::= (TYPE IN KEYWORD + formulate query(**B2**)) | (select item(**B3**) + formulate query(**B2**))

B3 select item ::= CLICK PULL DOWN MENU + CLICK ITEM

B4 navigate results ::= (browse results(**B5**) + navigate results(**B4**)) | (CLICK NEW SEARCH LINK + information seeking(**B1**))

B5 browse results ::= (show results(**B6**)-OBSERVE RESULTS + browse results(**B5**)) | (show results(**B6**)-COUNT RESULTS + browse results(**B5**)) | (CLICK ITEM + browse results(**B5**)) | (CLICK SORTING LINK + browse results(**B5**))

B6 show results ::= CLICK SIDEBAR | (CLICK SIDEBAR + CLICK NEXT PAGE LINK)

The number of rules and options within rules reflects the interactive nature and number of alternative choices provided by these two interfaces. Note that we used the terminals such as CLICK SEARCH BUTTON and CLICK VISUAL BAR which strictly speaking are not the lowest level of user behaviors, however, using higher level abstraction as terminals is suitable for interactive display-based systems [4] and ensures later data analysis. Many rules are defined recursively and consist of several options, which essentially reflect the interactivity of the graphical user interface (GUI). For example, a fairly interactive user behavior in RB++, “browse results (A6)”, consists of either ‘OBSERVE RESULTS’, ‘CLICK RESULT ITEM’, ‘CLICK SORTING BUTTON’, explore results, or any combination of the above.

From the BNF definition, we can see that RB++ is a more interactive interface than the baseline because it involves more rules and recursive definitions. However, it is not necessarily a complicated interface, since the rules for the RB++ interface are largely composed of a set of options instead of a sequence of user behaviors, which means that many rules are not executed for some types of tasks. Based on the BNF grammars, we hypothesize that for the simple search tasks, the RB++ interface will not necessarily be significantly different from the baseline interface, but for complicated searching and browsing tasks, that require more interaction or collection exploration, the RB++ will be significantly more effective, efficient, and satisfying than the baseline. For simple look up type tasks, both interfaces involve the sequence of user actions: formulate query, CLICK SEARCH BUTTON, and navigate results (see rule A1 and B1). Navigation of results is simple for this type of task in that it only involves the judgment of zero or non-zero results, which is trivial in both interfaces. Formulation of the query in this case involves typing in keywords and/or selecting the items from the interfaces (see rule A2, A4 and B2, B3). Even though item selection in the baseline interface involves two clicks (see rule B3) which means a slightly longer time to execute than in RB++, which only needs one click on the visual bar for item selection (see rule A4), we expected no significant difference. For type 2 tasks that involve data exploration and analysis, interaction with the visual bars of the RB++ interface provides an effective and efficient interaction style. Two typical sequences of user behaviors to complete type 2 tasks are: explore the collection by clicking (or mousing over) and observing the visual bars (see rule A1 and A3), or formulate a query and then explore the results by observing the visual bars (see rule A1, A5, A6, A8 and A9). With the traditional interface to finish type 2 tasks, users have to formulate a query and then literally scan and count all the results (see rule B1, B4, B5, and B6), which is time consuming.

We also hypothesized that users would exhibit rich interaction during their navigation of the results with RB++ (see rule A5 to A10). Actions of typing in keywords and clicking visual bars to filter results (rule A10) would be used frequently and interchangeably by the users to finish complex search tasks, especially when large numbers of results are returned.

3. RESULTS

3.1 Phase One Results

Table 1 lists the average time (in seconds) across all the participants to finish tasks 1 to 9 using the two different interfaces. Notice that we allowed the participants to stop the task if they felt that the task was too hard or too time-consuming to finish. It turned out that there were five participants who stopped task 5 and eight participants stopped task 6 before completion when they used the FilmFinder. Performance data of these participants were discarded for the unfinished tasks.

Table 1. Performance data (in seconds)

Task	**1 (.879)**	**2 (.522)**	**3 (.026)**	**4 (.000)**	**5 (.000)**
RB++	14.4	16.1	17.0	18.9	15.7
FilmFinder	14.7	14.4	29.7	40.7	204.0
Task	**6 (.000)**	**7 (.000)**	**8 (.000)**	**9 (.000)**	**10**
RB++	12.7	13.5	27.1	20.6	N/A
FilmFinder	328.0	87.2	101.3	112.8	N/A

Paired sample t tests on the performance data were computed and the p values are shown in the parenthesis for each task. We can see that except for the first two tasks (which were type 1 tasks), the performance differences between the two interfaces were all statistically significant at the .05 level. Clearly, RB++ supported superior performance for type 2 tasks.

We also counted error rates for tasks 1 to 9, which are listed in Table 2. The error rate was calculated as the number of participants who gave the wrong answer to the task divided by the total number of participants. We can see that except for the 8th task, no participants got wrong answers for any of the tasks using the RB++ interface. The error rates of the baseline interface were much higher than that of the RB++ interface, especially for tasks 5, 6, and 7. Notice that we did not consider those participants who gave up the task 5 or 6 using the Filmfinder, so the actual denominators used for calculating the error rates for these tasks were smaller than the total number of participants.

Table 2. Error rates

Task	**1**	**2**	**3**	**4**	**5**
RB++	0/15	0/15	0/15	0/15	0/15
FilmFinder	0/15	0/15	0/15	2/15	5/10
Task	**6**	**7**	**8**	**9**	**10**

RB++	0/15	0/15	1/15	0/15	N/A
FilmFinder	4/7	13/15	2/15	5/15	N/A

We also did paired sample t tests on the three satisfaction questions[2] which were completed after each task. Each response was given on a 5 point scale from strongly agree (5) to strongly disagree (1). For the first three tasks (simple lookups), there were no statistically significant differences between the two interfaces on any of the 3 questions. On the exploratory tasks (4-9), statistically significant differences favoring the RB++ were found on all three of the satisfaction questions.

We also compared the results for the seven overall usability questions on each interface asked after participants had done the tasks with each interface. Their responses were also given on a five point scale from strongly agree to strongly disagree. In each of the seven ratings, statistically significant differences were found favoring the RB+ interface. Clearly, satisfaction with the RB++ was greater than that with the Filmfinder.

There were also open-ended questions that the participants answered after finishing both interfaces. All of the participants considered the RB++ interface to be easier to use, especially for the complex searches. They commented on the easy use of the visual display with the multiple categories, which made it easy to combine the search criteria and narrow down the data, and they also thought it was good to be able to manipulate the search results in multiple ways. Thirteen out of 15 participants indicated that the RB++ interface gave them more confidence to complete the tasks. It was easy to go back and forth and to verify the results and the informative overview panel gave the participants more confidence to finish tasks. There was one participant who thought that both interfaces gave equal confidence and there was one participant who thought that the Filmfinder interface gave more confidence since he was more familiar with the Filmfinder and he felt somewhat confused by the dynamic feature of the RB++, but he acknowledged the usefulness of the dynamic feature in narrowing the results in the results panel.

When asked which interface better helped them gain an understanding of the library movie collection, the RB++ interface was chosen by all the participants. Again the visual display of the multiple categories and the cross reference of these categories was considered to be useful features for them to understand the whole collection. In addition, 10 out of the 15 participants indicated that they were more likely to use the RB++ interface if both were available. Three participants chose both interfaces, depending on the type of tasks, and two participants chose the FilmFinder because of its familiarity and aesthetic appeal.

For the question on the best thing about the RB++, participants pointed out the visual display of the multiple categories, its cross reference ability, the dynamic matching ability of the searching boxes and the one screen display of the results as opposed to the multiple page display of results in Filmfinder. As the worst thing about the RB++, participants indicated that it was not as aesthetically appealing as the Filmfinder and not quite as intuitive to use as the Filmfinder. Two participants specifically mentioned that the constant changing and updating of the interface made it a bit confusing.

[2] It is easy to use the interface
I feel satisfied with the results I got
I feel confident with the results I got

3.2 Phase Two Results

During the second phase we also asked participants to fill out the satisfaction questionnaires after finishing each task and these ratings were predictably high (all means above 3.5) More importantly, participants were also required to answer a set of open-ended questions after finishing the second phase. For the first question: "What is your overall impression of this interface for finding the statistical data?" the overall impression was positive. Participants used phrases such as "fairly easy to use", "very helpful in finding the information", "good for quick searching". There were also a couple of negative comments such as: "interface still came up with many results after filtering", "title of the results are not descriptive enough". Only one participant said that he did not like the interface, because of the poor categorization of information items under some categories which made him frustrated.

When answering the second question: "Was it helpful to understanding what is available at EIA?" all the participants thought the interface was helpful in that regard, which was largely attributed to the visual display of the categories, which gave them a sense of what the website covered. One participant wished that there were more categories displayed. The questionnaire also asked if the search boxes were helpful in completing the tasks. Participants gave high praise to this feature with comments such as "it's great to be taken directly to the page but not to have your results lost", "I like the way it narrows the focus and sort of guides a person to the info sought", "I didn't have to be concerned with performing a complex search that may return a null set-the results reflected my search string instantly". Two participants also commented that the feature was somewhat limited in use since relevant information may not appear in the title, or description.

4. DISCUSSION

The results strongly support that the RB++ interface was more effective and efficient in completing type 2 tasks than the baseline interface and that users felt more confident and satisfied with the RB++ in completing type 2 tasks. The higher effectiveness, efficiency and satisfaction gained in the RB++ resulted mainly from two aspects: the visual display of the statistical summary of the information items and the dynamic keyword searching capability in the results panel. The visualization bars helped the users understand relative proportions of items at a glance and use the posting numbers directly, which is much faster than literally counting. If we look at the BNF grammar, completion of type 2 tasks in RB++ only required participants to explore the collection (see first option of rule A1) without submitting queries to the database and then observing and counting returned results, which are necessary steps for the baseline interface to complete the same tasks (see rule B1).

The dynamic search boxes allow users do further filtering based on certain criteria and give users feedback on the filtered results instantly and continuously, which not only encourages the users to use this function, but also improves their efficiency. Another interface feature: displaying all the results on one screen might also help improve the efficiency and satisfaction, as several users mentioned.

Several components were tightly coupled in the interface with displayed search results. The search boxes are tightly coupled with the results, which means that any input in the search boxes will invoke instant filtering on the results. The visual bars are tightly coupled with the results and as such they support two functions. One is that any operations on the visual bars such as mouse over and selection, invoke the instant filtering of the results. The other is that any update of the results also updates the summary statistics in the visualization on the bars. Coupling provides users more ways to interact with the system and make the interaction more natural and smooth (see rule A8, A9, A10), which suggests a different interaction style for finding information than traditional search interfaces which tend to require discrete, well-defined turn-taking between the user and system. Traditionally, when users get to the results page, all they can do is browse the results. If they want to refine the results, they have to go back to the search interface, type in the refined keywords, click the search, and browse the new results, which not only interrupts the normal results browsing interaction, but also loses the current result set. RB++ encourages users to get an initial manageable result set and then refine it using one interface window without the need to go back and forth. Instead of displaying a set of static results, RB++ offers an effective and efficient means for users to understand the results by displaying summary statistics bars which give both visual and numeric data (see rule A9), and to explore results by providing ways to dynamically and continuously filter (see rule A10). The result set can be as large as displaying the whole collection, or as small as only one item, which depends on the initial query on the collection. In the second phase study, most of the participants completed their search tasks without doing a second query on the initial interface. The study showed that participants could utilize the initial interface to get an initial result set by selecting relevant categories and then narrow down results and find relevant web pages by exploring the results set. Typing in keywords (or string patterns) in search boxes was found to be the most frequently used means to explore and filter result set. These features were highly appreciated by the participants as seen from their comments.

5. RELATED WORK

Many information access interfaces try to provide a starting point for users by presenting overviews of the collection [9]. Overviews can help users understand the whole collection and select or eliminate sources from consideration. Overviews can direct users to subcollections quickly, where they can explore the details. Usually two types of overviews are employed: category overview and graphic overview. The category approach of Yahoo is a good example for the category overview. The HiBrowse interface for viewing category labels hierarchically based on the facets is another example [14]. A more recent information access interface using the category overview by presenting faceted metadata is the Flamenco interface [21]. The last two interfaces not only present the category labels to the users but also inform the users of the number of documents under each category. However, these interfaces do not allow users to employ simple mouse moves to quickly explore the relationship between different categories (or facets). The Flamenco interface could do this as part of its browsing and searching efforts, but it requires many commitments from users such as clicking the category and waiting. The previous version of the relation browser [13] presented various categories and allowed users to explore the relations by mouse over operation, but the interface only allowed the users to mouse over the main category.

The graphic overview is another type of overview, which usually employs various information visualization techniques. Lin [12] used the Kohenen map to visually present a topical overview of the collection. Each block on the map represents a subcollection with similar topics which are labeled by one or two salient words extracted from the subcollection. The adjacency of blocks indicates the topic similarity between subcollections. Wise, et al. [19] developed a three dimensional interface to visually present various topics. Zhang, et al. [25] exacted the key concepts from a collection and visually presented the concepts in a spring-embedded graph. Similar concepts were clustered together and usually represented as subtopics. The graphical overview is visually appealing, but the usability of this kind of interface has yet to be explored. 3-D interfaces are more problematic than 2-D interface in terms of ease of use and learnability. It seems that textual labels of category structure are more understandable than graphical representation.

Some research has been conducted on how to present the retrieved results in context. Hearst [10] used clustering techniques to cluster retrieval results on the fly and presented different clusters with labeled words to the users to help them understand of the results. Chen, and Dumais [3] employed classification techniques to categorize retrieved results based on the existing category structure and displayed them in hierarchical categories. Zamir and Etzioni [22] developed an interface that used on-the-fly clustering of metasearch results. These interfaces cluster or categorize the retrieval results on the fly, so scaling is problematic. The RB++ categorizes the collection offline and uses a uniform category structure to present overviews of the collection and the retrieval results. Consequently, RB++ can be scaled up easily. However, because RB++ depends on the metadata to reside on the client side to achieve its dynamics, it also suffers a different kind of scalability limitation. To date, we have had good success and response with data sets with tens of thousands of records and a dozen or so facets, however data sets with millions of records and scores of facets are problematic.[3].

There also has been some work on fast location of specific information items. Sorting is a prevalent means to help users locate a specific item. However, users still need to visually go through a list of items. The Alphaslider [1] is a visual component to help users quickly locate a known string of items, but it’s not very easy to use, especially for novice users. Besides, The Alphaslider can only locate the information items based on the first letter alphabetically. RB++ provides an easy and flexible way to locate the information items by typing in string patterns and the patterns can be matched anywhere in the information items. A similar technique is actually used in some applications such as the address box of Internet Explorer

[3] Various RB++ examples are available at http://idl.ils.unc.edu/rave/examples.html

browser, but the patterns are limited to matching from the beginning of the query string.

Dynamic query was a new type of interface [16] that inspired the original relation browser work. The interface visually displays the information items and provides the visual controlling components to explore the information items by tightly coupling search and visual display of results. RB++ uses this design concept, but instead of providing a visual interface, RB++ employs a more understandable (especially for topical overview) category structure for the information items. Moreover, the search box is a very effective and efficient component for the non categorized attributes of the items, while the visual controlling components such as sliders or check boxes can only be used for controlling categorical attributes of the items.

Query preview [18], attribute explorer [17] and other interfaces [11], and [20] provide similar ways to explore the relationships between different facets of the classification. These interfaces worked for structured information such as that found in databases. Of course, all these types of interfaces depend on good underlying categorization of data. Our long-term goal is to make the interface work for unstructured textual information. The search boxes provided are a first step in this direction, although they are currently limited to search within the fields specified in the results display.

6. LIMITATIONS OF RB++ AND ONGOING WORK

One constraint in RB++ is the limited number of categories that can be displayed, which is affected by two factors. One factor is screen real estate. We can partially alleviate the issue of screen real estate by utilizing a Zoomable User Interface (ZUI) to display the categories. We have experimented with integrating the Jazz toolkit [2] into the interface and this provides aproximately a ten-fold increase in the number of facet values that can be supported within each facet, although at the expense of some of the mouseover dynamics since the mouse must now be used for zooming as well as normal hovering. Another factor is size of the memory to hold the client-side distribution counts data, the number of which increases exponentially with increased number of displayed categories. One way to solve the issue is to only calculate part of the distribution counts data, which hopefully are most frequently used during the user's interaction with the interface. Other approaches, such as employing novel data structures were also suggested by Doan, et al. [5]. However, all these solutions have to sacrifice the interactivity of the interface that depends on client-side metadata to support rapid mouse activities. For example, preloading partial distribution counts data for large numbers of categories make some distributional data and visualization unavailable when users try to re-partition the information space by mouse moves. At present, the best development path seems to be hierarchical partitioning of very large information spaces with multiple RB++ instances that require a new download for each of the cascading subsets.

Another constraint of RB++ is the limited matching function of the search boxes. The interface currently matches input string patterns to the corresponding result fields on the lexical level. Matching in this level is sufficient in many cases such as matching with fields with numbers or short textual strings such as titles, but for the fields with more semantic bearing strings such as descriptions of web pages, a more sophisticated match function based on semantics might be needed—perhaps a kind of full-text engine in each text field, although the close coupling with the facet panel will then be in doubt.

Currently, the interface provides a uniform category structure for both the entire collection and the retrieved results set. This is good for its consistency. However, for the retrieved result set, a more fine-grained category structure might be better for users to understand it and conduct string searches.

Overall, the RB++ represents an example of a highly interactive user interface that offers improved performance and satisfaction for users of large websites and digital libraries. It can find application as the entry point for a large website or as a way to work with large results sets returned from search engines as long as the data is structured in advance.

7. WORK ON AUTOMATIC CLUSTERING AND CLASSIFICATION

Over the past few years we have created more than two dozen instances of data sets using RB and RB++, demonstrating its applicability as an interface to many different types of data. If the data resides in a database, it is possible to map the scheme to the underlying RB++ scheme (see [24] for details on the system architecture) and simply import the data automatically. For many WWW applications, this is not possible so we have been developing ways to automate facet discovery and webpage assignment to those facets. The basic approach is to crawl the website(s), create term-document representations and then use machine learning techniques to cluster the webpages and extract candidate labels, use human judgment to select the best labels, and then classify the webpages into those categories using the statistical model produced in the process. See Efron et al [6,7] for details of the techniques we have applied to date.

To illustrate the current state of development, consider the EIA example used in the study reported here. The categories displayed on the EIA RB++ instance were originally created manually, which certainly did not scale well to many other large information collections such as various other government statistical web sites. Figure 7 is a screen shot of the RB++ instance for the Bureau of Labor Statistics web site that uses topic facets and webpage assignment that were automatically determined. About 13,000 HTML web pages were crawled from web site and a soft clustering method was then applied to those pages. For each webpage, the statistical model yielded a probability of belonging to every cluster. Thus, every page 'belongs' to every cluster at some level of probability. The first topic column contains all the pages with highest probability of belonging to those facet values. The second column contains all the pages with the second highest probability values for those facet values. The third and fourth columns are the months and years of page update extracted from web pages themselves—facets that are know to be problematic but used for illustration since our primary emphasis was on topic discovery. The display of two topical columns reflects the underlying characteristic of soft clustering where items can appear in multiple clusters. However, our discussions with BLS staff and demonstrations to other potential users show that this two-column display is confusing. Therefore, only the first topical column was kept in a later version of the RB++ BLS instance (see figure 8). An additional facet: geographical coverage, which is an important facet of government statistical web sites, was added in this version. The assignments were made using a rule based classification method to classify the web pages into categories of various geographical coverages.

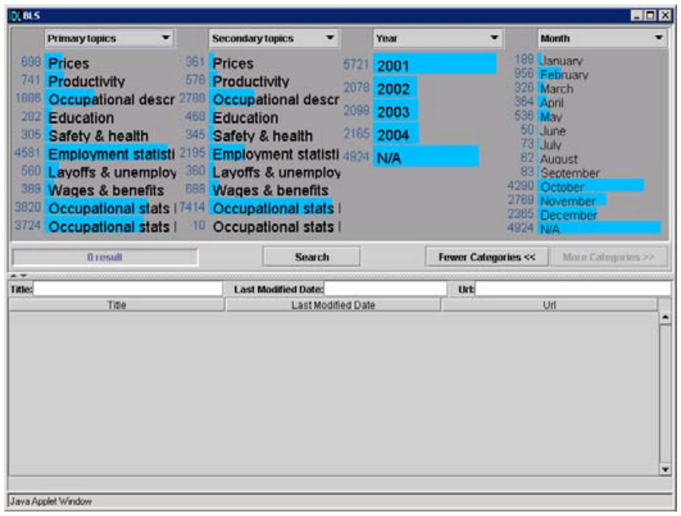


Figure 7. First version of RB instance for BLS web site

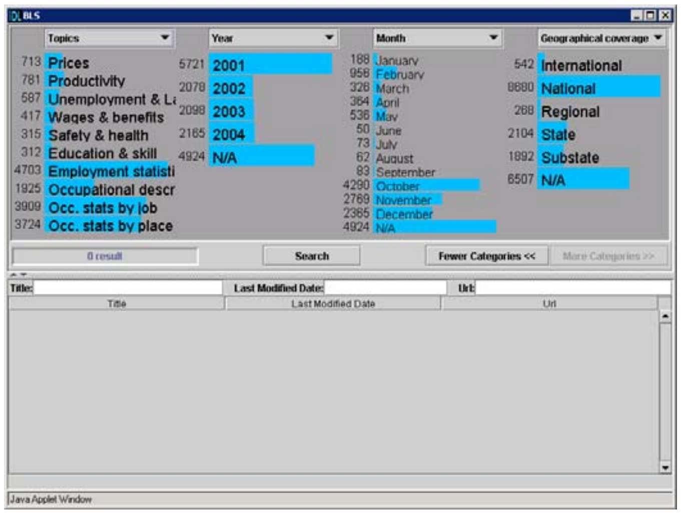


Figure 8. Latest version of RB instance for BLS web site.

In addition to the BLS instance, we have used these techniques to create RB++ instances for the FedStats website and are working to create instances for other federal statistical agencies. At present, the FedStats and BLS instances are installed on FedStats servers and are being tested by federal statistical agency personnel. We also hope to address the important facet of time coverage of data itself in future work.

Overall, the RB++ user interface has evolved to a state where it can be easily applied to many kinds of well-structured data. The user testing reported here demonstrates the efficacy of the interface for search and browse tasks that would be very difficult to execute with SQL syntax or form fillin interfaces. Our current efforts are to develop techniques for automatically populating the RB++ database with unstructured data from the WWW. To date, these efforts have led to promising prototypes for several federal statistical websites.

8. ACKNOWLEDGMENTS

This work was supported by NSF Grant EIA 0131824. The authors wish to thank Paul Solomon and other anonymous reviewers for their valuable comments on the paper and Tim Shearer for designing the underlying database structure of the interface. We also thank Jonathan Elsas and Miles Efron for developing the clustering software and techniques.

9. REFERENCES

[1] Ahlberg, C., and Shneiderman, B. The alphaslider: a compact and rapid selector. In *Proceedings of the SIGCHI conference on human factors in computing systems*. Boston, Massachusetts. 1994.

[2] Bederson, B., Meyer, J., and Good, L. Jazz:An Extensible zoomable user interface graphics toolkit in java. In *ACM UIST2000*, 171-180.

[3] Chen, H, and Dumais, S. Bringing order to the web: Automatically categorizing searching results. In *Proceedings of the SIGCHI conference on human factors in computing systems*. The Hague, Amsterdam. 2000

[4] Dix, A., Finlay, J. Abowd, G. Beale, R, and Finley, J. *Human-Computer Interaction* (2nd Ed.). Prentice Hall, Hillsdale, NJ, 1998

[5] Doan, K., Plaisant, C., Shneiderman, B., and Bruns, T. *Interface and Data Architecture for Query Preview in Networked Information Systems*. ACM Transactions on Information Systems, July 1999, Vol. 17, No. 3, 320-341.

[6] Efron, M., Marchionini, G. and Zhang, J. Implications of the recursive representation problem for automatic concept identification in on-line governmental information. *ASIST SIG-CR Workshop*, Long Beach, CA, 2003.

[7] Efron, M., Elsas, J., Marchionini, G., and Zhang J.. Machine learning for information architecture in a large governmental website. Joint Conference on Digital Libraries 2004 (Tuscon, AZ, June 7-11, 2004)

[8] Greene, S., Marchionini, G., Plaisant, C., & Shneiderman, B. (2000). Previews and overviews in digital libraries: Designing surrogates to support visual information seeking. *Journal of the American Society for Information Science*, 51(4), 380-393.

[9] Hearst, M. User interfaces and visualization. In *Modern information retrieval*. Ed. by Baeza-Yates, R., and Ribeiro-Neto, B. Chapter 10, ACM Press, New York, NY, 1999 257-324.

[10] Hearst, M. and Pedersen, P. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results, *Proceedings of 19th Annual International ACM/SIGIR Conference*, Zurich, 1996

[11] Lanning, T., Wittenburg, K., Heinrichs, M., Fyock, C., and Li, G. Multidimensional information visualization through sliding rods. AVI'02 Palermo, Italy.

[12] Lin, X. Map displays for information retrieval. *Journal of the American society for information science*. 1997 48(1), 40-54.

[13] Marchionini, G., and Brunk, B. Toward a general relation browser: A GUI for information architects. *Journal of Digital Information*. Article No. 179, 2003-04-09 2003 4(1). http://jodi.ecs.soton.ac.uk/Articles/v04/i01/Marchionini/

[14] Pollitt A. S., Ellis G. P., and Smith M. P. HIBROWSE for Bibliographic Databases *Journal of Information Science*, 1994 20 (6), 413-426.

[15] Reisner, P., Formal Grammar and human factor design of an interactive graphics system. *IEEE Trans. on Software Engineering*, 7(2), 229-240, 1981

[16] Shneiderman, B., *Dynamic queries for visual information seeking*, IEEE Software 11, 6 (1994), 70-77.

[17] Spence, R, and Tweedie, L. The attribute explore: information synthesis via exploration. *Interacting with Computers*. 1998 11, 137-146.

[18] Tanin, E., Lotem, A., Haddadin, I., Shneiderman, B., Plaisant, C., and Slaughter, L. Facilitating data exploration with query previews: a study of user performance and preference. *Behaviour & information technology*. 2000 19(6). 393-403.

[19] Wise, J., Thomas, J., Pennock, K., Lantrip, D., Pottier, M. and Schur, A. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Proc. of the Information visualization Symposium 95*, pages 51-58. IEEE Computer Society Press, 1995

[20] Wittenburg, K., Lanning, T., Heinrichs, M., and Stanton, M. Paraell bargrams for consumer-based information exploration and choice. UIST' 01,Orlando FL.

[21] Yee, K, Swearingen, K, Li, K, and Hearst, M. Faceted metadata for image search and browsing. CHI'03, Ft. Lauderdale, FL.

[22] Zamir, O. & Etzioni, O. (1999). Grouper: A Dynamic Clustering Interface to Web Search Results (WWW8 1999) http://www8.org/w8-papers/3a-search-query/dynamic/dynamic.html

[23] Zhang, j. Relation browser++: An interface for exploring and searching large information collections. *Digital government conference 2004.* Seattle. WA 2004.

[24] Zhang, J., and Marchionini, G. *Relation Browser++: a Fast and Contextualized Searching and Browsing Tool.* Technical Report SILS-TR-2004-01. University of North Carolina at Chapel Hill, 2004.

[25] Zhang, J., Mostafa, J., and Tripath, H. Information retrieval by semantic analysis and visualization of concept space for the D-lib magazine. *D-lib online magazine*, 2002 (10).