# The future is in the past: Designing for exploratory search

Gene Golovchinsky[1], Abdigani Diriye[2], and Tony Dunnigan[1]
[1]FX Palo Alto Laboratory, Inc., Palo Alto, CA, USA
[2]University College London Interaction Centre, University College London, UK
{gene, tonyd}@fxpal.com, a.diriye@ucl.ac.uk

## ABSTRACT

Exploratory search activities tend to span multiple sessions and involve finding, analyzing and evaluating information found through many queries. Typical search systems, on the other hand, are designed to support single query, precision-oriented search tasks. We describe a search interface and system design of a multi-session exploratory search system, discuss design challenges encountered, and chronicle the evolution of our design. Our design describes novel displays for visualizing retrieval history information, and introduces ambient displays and persuasive elements to interactive information retrieval.

## Categories and Subject Descriptors
H5.3. Information interfaces and presentation: Group and organizational work.

## General Terms
Search interface, interface design.

## Keywords
Multi-session search, collaborative search, exploratory search, search interface, ambient displays, persuasive computing.

## 1. INTRODUCTION

Online search has become a big part of people's daily information seeking tasks. Many of the tasks carried out online can be classified as being known-item vs. exploratory search. Known-item searches are concerned with finding a single document, factoid, or snippet that satisfies the person's information need. Such interactions are typically characterized by short-duration, simple queries to search systems, and by the examination of only the top few search results retrieved. This kind of activity is well-supported by major web search engines.

Exploratory search tasks, on the other hand, may involve many queries, many retrieved documents, may include activities of multiple people, and may evolve over time. Such information seeking occurs commonly in medical, legal, pharmaceutical, intelligence, and academic fields, in addition to such consumer activities as travel planning or personal health research.

One key characteristic of these kinds of information seeking tasks is the large number of queries that may be run to find the needed information. Many queries are required for several reasons: to gain a better understanding of the topic, to examine independent aspects, to react to newly-found information, etc. In many cases, these queries return some of the same documents. Yet most search engines treat each query in isolation, forcing searchers to rely on their memory to make sense of the newly-found results.

Based on these observations, we set out to build a tool for exploratory search that makes histories of people's search activity available to them explicitly. In this paper, we describe our decisions and the process through which we arrived at the current design. We focus specifically on the role of history of interaction with the system to help people make sense of their ongoing activity. We describe the evolution of both the interface design and the system architecture, as design tradeoffs between these components have a direct impact on the user experience. We then sketch out some ideas for introducing ambient displays of information into the information seeking interface.

This work makes the following contributions: it describes the evolving design of a complex interface for information seeking; it introduces novel interactive components for visualizing session state; it describes the use of process-oriented metadata for managing results in addition to conventional document-oriented metadata; and it proposes the use of ambient and persuasive displays to provide feedback in information seeking tasks.

## 2. BACKGROUND

Many models of information seeking have been proposed over the years; see White and Roth [31] for a recent summary. Such models represent a range of behaviors associated with information seeking, including identifying sources, running queries, examining results, reformulating the information need, etc. Models also capture the contexts of information seeking activity, including motivating tasks, emotional states, and organizational, social, cultural factors [19]. The vast majority of these models represent information seeking as an interactive, evolving, learning behavior.

Yet the majority of systems through which people search for information treat each query as an isolated event [22]. While web search systems certainly keep track of a user's ongoing behavior to improve query precision of future similar queries, no information is reflected back to help with the searcher's current task. Behavioral data checks in, but it doesn't check out.

Much of the emphasis—both in research and practical settings—has been on improving ranking efficiency and effectiveness for single-shot single-user search. But in exploratory search, the queries are not independent. They represent a conversation with the system through which searchers try to understand, express, and evolve their information need [6]. When it comes to exploratory information seeking tasks, searchers have been left pretty much to their own devices.

Multi-session search tasks comprise a significant portion of most web search activity [21][28][25]; searchers undertaking multi-session search tasks tend to make heavy use of bookmarking features, re-searching to relocate previously seen content, and printing, emailing and saving pages to tackle these search tasks [5]. A number of features exist within experimental search systems and browsers to support these categories of search tasks, although as studies (e.g., [25][11]) showed, such features are rarely found in the wild, where people typically rely on bookmarks and email to manage their found information.

The ability to keep track of queries in a search session extends to the early days of DIALOG and MEDLARS, which allowed queries to be re-run and combined via Boolean operators. While this interface feature made it possible to review what was done and to save some re-typing, it did not help in managing the overall set of documents retrieved for a given search task.

Early experimental systems such as Ariadne [30] made it possible to review a search session after the fact to get a retrospective sense of the distribution of activities between top-level menu selections, search commands, and the examination of specific search results. This interaction was captured only as a series of screenshots that facilitated review without the ability to interact with the earlier states of the system. C-TORI [18] made it possible to keep track of and share queries in a search session, but did not allow reasoning about the retrieved documents.

Ahn *et al.* [2] integrated some cues from group use of documents into the display of search results, showing that aggregations of prior use of documents could be useful. This system relied on aggregating user behavior over some length of time; this is quite different from aggregating results over successive queries in the same session, independent of any document's retrieval outside the given search task.

SearchPad [8] and SearchBar [26] kept track of users' queries issued, visited URLs, and saved documents, but did not help assess or manage the overlap in results among the various queries run during a session. Session Highlights is an example of a bookmarking tool that saved URLs [20] to a repository to help manage revisiting of documents; it did not, however, have any specific awareness of the structure of search activity.

In designing Querium, we wanted to explore the design space more fully, to look at how the retrieval histories of individual documents (and not just sequential query lists) could be used to aid information seeking. While earlier systems kept track of some aspects of the search history, they did not provide systematic means to help people reflect on what was found, and to use that information to plan subsequent search moves.

In the remainder of the paper, we document the design issues encountered when constructing a multi-session exploratory search system. We begin with the search interface and system design, describe a redesign based on an informal evaluation, and conclude with a discussion of future directions.

## 3. QUERIUM 1.0

We designed Querium to explore a range of topics related to interactive information retrieval. We chose to emphasize interactivity over incremental improvements in retrieval performance, and focused on real tasks rather than simulated information needs. The focus on exploratory search drove some of the design decisions we outline below.

The emphasis on real tasks and exploratory search constrained our choice of document collections to index. The desire to study the use of relevance feedback made it necessary to maintain our own index. This, in turn, made it difficult to search the open web. After considering Medline and patent document collections, we settled on CiteSeer [13], a collection of scientific papers from a variety of disciplines, harvested from the open web. The CiteSeer snapshot we are using contains more than 1.6 million unique academic papers, consisting of more than 100GB of text. This provides us with a sufficiently large collection to support real information seeking tasks. Furthermore, this is a collection that is used by thousands of people in academia and related fields for conducting literature surveys and other research activity, which should allow us to recruit participants to perform real information seeking activity rather than relying on synthetic tasks.

Golovchinsky and Pickens [17] proposed a model of propagating context among information retrieval objects in the course of an interactive session. That paper described how contextual information could flow from documents to queries, from one query to a set of queries, from queries to links, etc. Our design process started with the naïve implementation of several such context flows, including several ways to select documents for relevance feedback, and a mechanism for merging results of multiple queries. As we built and analyzed various designs, we discovered that careful attention was required to make these useful transformations comprehensible to users.

The main focus of this paper is an exploration of the use of historical (search session) metadata accumulated during the search process to help inform the evolution of the search activity. We want to understand the possibilities of using not only a record of the queries, but also of the retrieved documents, to help searchers explore a particular topic.

### 3.1 System Architecture

Querium implements a client-server architecture. The web-based client manages interactions with the server and displays the retrieved documents. The server consists of several components, as shown in Figure 1. The application server manages a session state database that records topics, queries, retrieved documents, assessments of usefulness, comments, etc. This information makes up the history of interaction with the system.

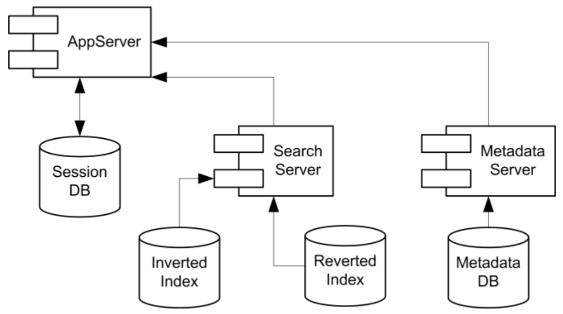


**Figure 1. Querium architecture**

Document retrieval is handled by a separate search service that manages two indexes: an inverted index for document content, and a secondary "reverted" index [27] that is used for relevance feedback. Lucene is used to manage both indexes.

A second relational database contains metadata associated with the indexed documents, including author, title, venue, and publication date information. When a document is retrieved for the first time, a new record is created for it in the session database, and some of the associated metadata are transferred to the session database.

The session database schema consists of Topic instances that organize search tasks; each topic is owned by some person and may have additional members. Queries are organized by Topic. Each Query includes either some text or one or more relevance feedback documents (or both), and has an associated list of Postings. A Posting associates a Document with a rank with respect to some Query, and contains the best snippet from the matching document. The database also contains comments associated with documents.

## 3.2 Search User Interface

Our initial design (Figure 8) included several important components, which we describe below. The first few items below provide useful functionality for expressing information needs and for assessing the results; the rest—the query history, retrieval histograms, faceted filters, and the summary view—deal directly with process metadata. In subsequent sections, we show how we evolved the design of these components based on user feedback.

### *3.2.1 Queries*

Querium offers three types of queries: standard keyword queries, relevance feedback, and document fusion. Relevance feedback could be initiated in two ways: by using “liked” documents or by selecting document *ad hoc* via the check-boxes on the left of each result. Fusion queries represent the combination of results of all queries in the task; documents can be ordered based on their metadata, by the time they were last seen, or based on a combination of their retrieval ranks in the session.

### *3.2.2 Integrated document view*

Previewing the document with the search results visible makes it easier to move seamlessly among the retrieved documents and to assess documents. Typical strategies people use for examining search results include opening documents in different tabs or windows. Opening documents in different tabs makes it tedious to move back and forth between the results and documents; opening documents in a new window requires more screen space than is typically available. By juxtaposing the document view with the search results, we allow people to preview documents without losing their search context. This strategy becomes increasingly more effective as the widths of people’s monitors increase.

### *3.2.3 Summary view*

We introduced a summary view that gave an overview of activity in a currently-selected task. It showed the list of queries (arranged by time of last activity), and within each query, it showed actions such as document relevance judgments, sharing events, comments, etc. A collaborator could use it to review recent activity; in a single-user mode, it was intended to facilitate review prior to resuming a search task.

### *3.2.4 Document assessment*

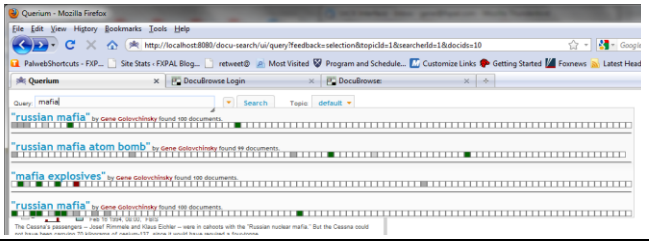
Querium allows people to mark documents with “like” / “dislike” controls to leverage people’s familiarity with this kind interaction; in addition, a sharing control was added to mark documents worthy of collaborators’ attention. This redundancy was used to probe perceptions of communication vs. search tools.

Documents start out being marked as “not seen” and the system will automatically mark them as “viewed” when the user browses the corresponding document. Documents marked as “not liked” are suppressed from further display. A filter (see below) can be used to reveal these documents on demand.

### *3.2.5 Query history*

A query history can be useful as a navigation aid in recreating an earlier state, to understand the totality of what has been done, or for making sense of what your collaborators have been doing. We used a type-ahead mechanism to show a subset of the query history that matched the currently-entered keywords to help people make sense of what they—or their collaborators—had searched on before. The full query history was also available from a dropdown menu adjacent to the query text box.

For each query, we displayed a list of color-coded squares, one per retrieved document, arranged in the order of retrieval. Color represents the assessment of each document—useful, not useful, seen, or not seen. This display (Figure 2) shows at a glance how much of a query’s search results had been explored, and how much potential overlap there was among queries. Clicking on a query loads the corresponding result set.


Querium - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/docu-search/ui/query?feedback=selection&topicId=1&searcherId=1&docids=10
PalwebShortcuts - FXP... Site Stats ‹ FXPAL Blog... retweet® Most Visited Program and Schedule... Customize Links Getting Started Foxnews Latest Head
Querium
DocuBrowse Login
DocuBrowse:
Query: mafia
Search
Topic: default

"russian mafia" by Gene Golovchinsky found 100 documents.
"russian mafia atom bomb" by Gene Golovchinsky found 99 documents.
"mafia explosives" by Gene Golovchinsky found 100 documents.
"russian mafia" by Gene Golovchinsky found 100 documents.
Feb 15 1994, 08:00, FBIS
The Cessna's passengers -- Josef Rimmele and Klaus Eichler -- were in cahoots with the "Russian nuclear mafia." But the Cessna could

**Figure 2. Query history drop-down from an earlier prototype.**

### *3.2.6 Retrieval histogram*

A histogram is displayed with every document in a result list [16]. The histogram shows the retrieval history of the associated document in the current task: each bar represents the results of one query; bars are arranged left-to-right according to the sequence of queries in the task. The taller the bar, the better the document matched that query. Each bar is linked to the corresponding results list. The figures below show some useful patterns: Figure 3 represents a newly-retrieved document, Figure 4 represents a document that is being consistently retrieved at high ranks, and Figure 5 shows a document that had previously been retrieved at a low rank (and probably not seen), but is ranked high with respect to the current query.

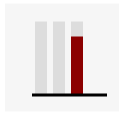


**Figure 3. Newly-retrieved document.**

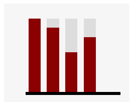


**Figure 4. Consistently-important document.**

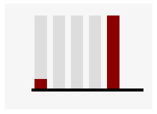


**Figure 5. Previously-retrieved but now important document.**

**Three distinct histogram patterns.**

These histograms are intended to give people a quick impression about the retrieval history of a particular document, and thus its significance with respect to the current task goals. If the goal is to find new documents, the user would look for patterns similar to Figure 3 or Figure 5; if the goal is to identify central themes, documents such as that shown in Figure 4 would become the focus of a searcher’s attention.

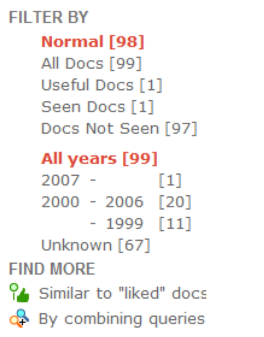


**Figure 6. Search sidebar with filters and query tools.**

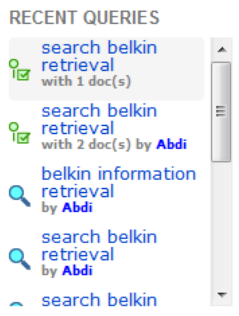

RECENT QUERIES
search belkin retrieval with 1 doc(s)
search belkin retrieval with 2 doc(s) by Abdi
belkin information retrieval by Abdi
search belkin retrieval by Abdi
search belkin

**Figure 7. Recent queries in the side bar.**

### *3.2.7 Faceted Filters*

Querium extended the notion of *document metadata* to also include *process metadata* (Figure 6). Thus in addition to the year of publication, we include process metadata that indicate how many useful, seen, and new documents were retrieved by the current query. The difference between 'Normal' and 'All Docs' is that the latter includes documents judged as 'Not useful', that are suppressed in other filter settings. The ability to filter on process decisions allows searchers to reason about what they've seen, what they've found useful, and what they've yet to explore. This is a key extension over earlier session-tracking systems such as SearchPad [8] that only echo back the history.

The various query expansion and fusion queries were grouped in a "FIND MORE" section of the sidebar (Figure 6); a history of recent queries was shown in the sidebar as well (Figure 7).

## 3.3 Evaluation

We performed ongoing evaluations of our design using a variety of heuristic analyses. These included design reviews by the authors, informal evaluations with representative users, and a variety of engagements with user interface design and information retrieval experts. These evaluations identified inconsistent and confusing aspects of the design, and served as inspirations for iterative design.

We also evaluated Querium through a limited deployment to a small group of academics outside our organization, who were asked to use Querium for their own information needs. The goal was to test whether the interface was comprehensible to people and to get some impressions about the usefulness of the system. This was not a summative evaluation, but rather a set of probes to help guide our design process. Specifically, we were interested in understanding the extent to which query results would overlap, whether people would use the query history, and whether relevance feedback controls were usable.

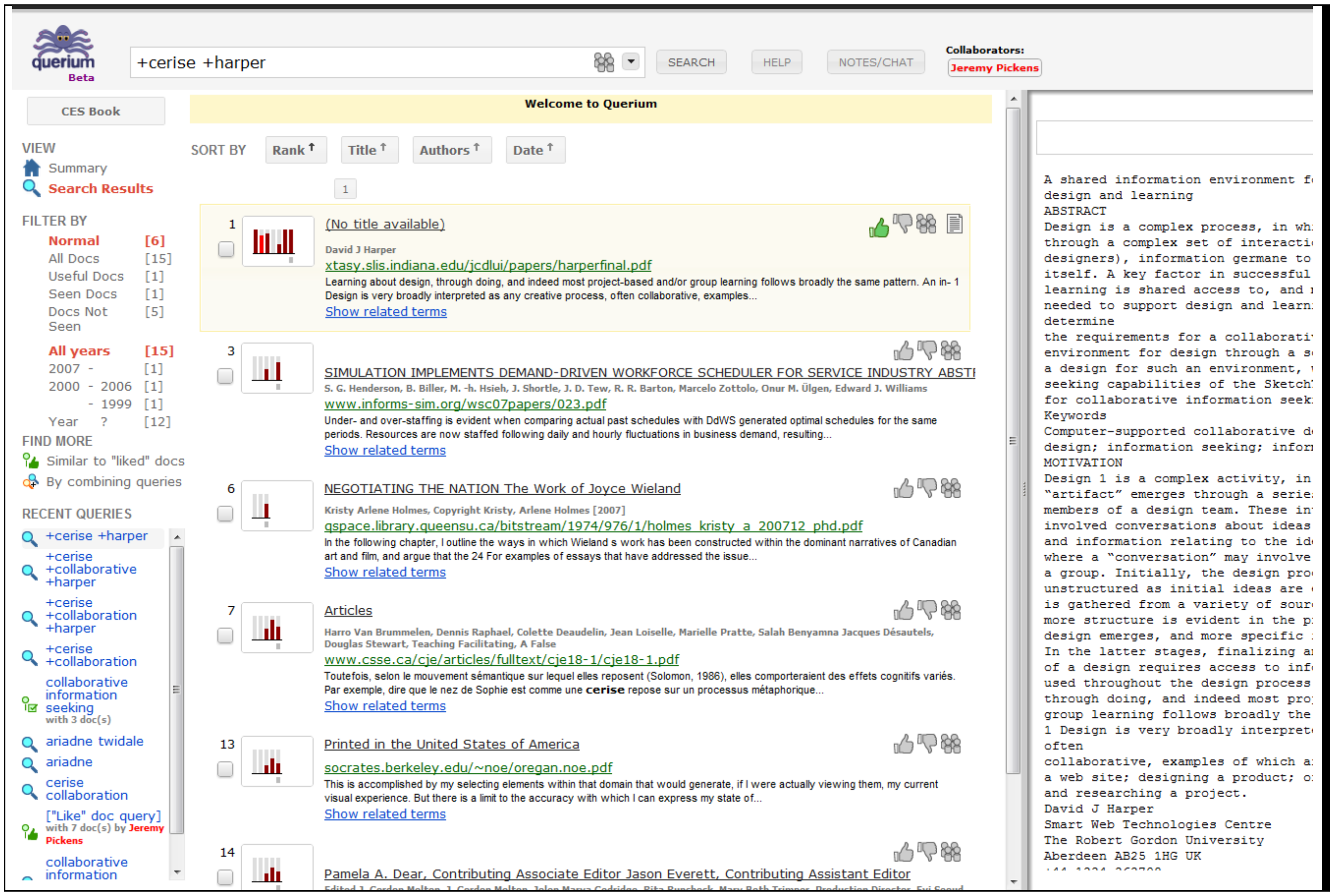
Participants were given a short video tutorial on the system, and were then asked to use it to pursue their own information needs. Nine participants performed 19 search tasks; many only created one, but some created as many as four separate tasks.

People's use of the system was recorded in search logs; we also interviewed them about their experiences with the system to understand which aspects of our redesign were successful, and which needed more work. Interviews were conducted in person and over Skype, depending on participants' locations. Interviews were transcribed, and responses were categorized to identify common trends.

### 3.3.1 *Log analysis*

We analyzed our usage log to help us understand patterns of query construction and revisiting. We divided system use by each participant into episodes consisting of at least 30 minutes of inactivity. Within each such episode, we counted the number of new queries and the number of revisits of prior queries. We also looked for other patterns querying within each episode.

The number of episodes varied from one to four per task ($\mu = 1.9, \sigma = 1.0$); most tasks had at least two episodes. In addition to the 94 queries that participants ran while using the system, they revisited earlier queries 55 times. Twenty six revisits came from retyping or reselecting the query from the query input field, six



1 (No title available)
David J Harper
xtasy.slis.indiana.edu/jcdlui/papers/harperfinal.pdf
Learning about design, through doing, and indeed most project-based and/or group learning follows broadly the same pattern. An in- 1 Design is very broadly interpreted as any creative process, often collaborative, examples...
Show related terms

3 SIMULATION IMPLEMENTS DEMAND-DRIVEN WORKFORCE SCHEDULER FOR SERVICE INDUSTRY ABSTI
S. G. Henderson, B. Biller, M. -h. Hsieh, J. Shortle, J. D. Tew, R. R. Barton, Marcelo Zottolo, Onur M. Ülgen, Edward J. Williams
www.informs-sim.org/wsc07papers/023.pdf
Under- and over-staffing is evident when comparing actual past schedules with DdWS generated optimal schedules for the same periods. Resources are now staffed following daily and hourly fluctuations in business demand, resulting...
Show related terms

6 NEGOTIATING THE NATION The Work of Joyce Wieland
Kristy Arlene Holmes, Copyright Kristy, Arlene Holmes [2007]
qspace.library.queensu.ca/bitstream/1974/976/1/holmes_kristy_a_200712_phd.pdf
In the following chapter, I outline the ways in which Wieland s work has been constructed within the dominant narratives of Canadian art and film, and argue that the 24 For examples of essays that have addressed the issue...
Show related terms

7 Articles
Harro Van Brummelen, Dennis Raphael, Colette Deaudelin, Jean Loiselle, Marielle Pratte, Salah Benyamna Jacques Désautels, Douglas Stewart, Teaching Facilitating, A False
www.csse.ca/cje/articles/fulltext/cje18-1/cje18-1.pdf
Toutefois, selon le mouvement sémantique sur lequel elles reposent (Solomon, 1986), elles comporteraient des effets cognitifs variés. Par exemple, dire que le nez de Sophie est comme une **cerise** repose sur un processus métaphorique...
Show related terms

13 Printed in the United States of America
socrates.berkeley.edu/~noe/oregan.noe.pdf
This is accomplished by my selecting elements within that domain that would generate, if I were actually viewing them, my current visual experience. But there is a limit to the accuracy with which I can express my state of...
Show related terms

14 Pamela A. Dear, Contributing Associate Editor Jason Everett, Contributing Assistant Editor

```
A shared information environment f
design and learning
ABSTRACT
Design is a complex process, in wh
through a complex set of interacti
designers), information germane to
itself. A key factor in successful
learning is shared access to, and
needed to support design and learn
determine
the requirements for a collaborati
environment for design through a s
a design for such an environment,
seeking capabilities of the Sketch
for collaborative information seek
Keywords
Computer-supported collaborative d
design; information seeking; infor
MOTIVATION
Design 1 is a complex activity, in
"artifact" emerges through a serie
members of a design team. These in
involved conversations about ideas
and information relating to the id
where a "conversation" may involve
a group. Initially, the design pro
unstructured as initial ideas are
is gathered from a variety of sour
more structure is evident in the p
design emerges, and more specific
In the latter stages, finalizing a
of a design requires access to inf
used throughout the design process
through doing, and indeed most pro
group learning follows broadly the
1 Design is very broadly interpret
often
collaborative, examples of which a
a web site; designing a product; o
and researching a project.
David J Harper
Smart Web Technologies Centre
The Robert Gordon University
Aberdeen AB25 1HG UK
```

**Figure 8. Querium interface: top area for new queries, filter and query history sidebar on the left, matching documents in the middle, and the document view (shown in part) on the right.**

came from using the recent queries list at the bottom of the sidebar, and 23 came from the summary page. There was no strong difference of new query to revisiting an older one as a function of the number of episode, as shown in Table 1. This lack of differences in revisitation may be due to the episodes occurring too close to each other in time. A longer active task might reveal different patterns, as searchers reacquaint themselves with the task history. The data do reveal a consistent trend of revisiting about one query for every two new ones run.

**Table 1. New vs. revisited queries by episode. RF stands for relevance feedback, that is, finding similar documents.**

| episode # / freq | new typed | new RF | query box | recent | sum mary | total revisit |
|---|---|---|---|---|---|---|
| 1 / 18 | 57 | 1 | 16 | 2 | 11 | 29 |
| 2 / 11 | 20 | 4 | 7 | 2 | 7 | 16 |
| 3 / 5 | 12 | 2 | 2 | 1 | 0 | 3 |
| 4 / 2 | 4 | 0 | 0 | 0 | 2 | 2 |

Our participants ran many queries and retrieved, viewed, and saved many documents. Table 2 shows for each task, how many queries were run, how many unique documents were found by those queries, how many documents were viewed, "liked", and "disliked." We omit "shared" documents, as only three participants used that feature. We note a considerable disparity in the degree of use of the various assessments: while a few participants used the "like" and "dislike" buttons extensively, many did not.

**Table 2. Queries and assessments by topic**

| Task # | N Queries in task | N unique docs | Total docs | overlap | Viewed | liked | disliked |
|---|---|---|---|---|---|---|---|
| 4 | 14 | 913 | 1227 | 0.74 | 17 | 3 | 1 |
| 5 | 11 | 528 | 1112 | 0.48 |  | 23 | 30 |
| 7 | 4 | 34 | 40 | 0.85 | 2 |  | 1 |
| 8 | 6 | 126 | 153 | 0.82 | 5 |  |  |
| 9 | 3 | 211 | 300 | 0.70 | 1 |  | 1 |
| 10 | 2 | 196 | 300 | 0.65 |  |  |  |
| 11 | 3 | 208 | 208 | 1.00 | 1 |  |  |
| 12 | 6 | 476 | 598 | 0.80 |  |  |  |
| 15 | 4 | 279 | 326 | 0.86 | 1 | 11 | 2 |
| 19 | 3 | 274 | 300 | 0.91 | 3 | 3 | 5 |
| 21 | 7 | 659 | 695 | 0.95 | 5 | 5 | 4 |
| 23 | 5 | 445 | 600 | 0.74 | 4 |  | 1 |
| 24 | 9 | 633 | 700 | 0.90 | 5 |  | 5 |
| 25 | 1 | 100 | 100 | 1.00 |  |  | 4 |
| 26 | 4 | 337 | 399 | 0.84 | 3 | 8 | 1 |
| 28 | 2 | 187 | 200 | 0.94 |  | 2 |  |
| 29 | 1 | 100 | 100 | 1.00 | 4 | 4 |  |
| 30 | 1 | 100 | 100 | 1.00 |  | 3 |  |
| 31 | 8 | 478 | 700 | 0.68 | 4 | 1 | 12 |

Table 2 also highlights the high rate of overlap in the results of the queries our participants ran. This corroborates our assumptions about the distribution of results among queries on a given topic; more explicit hypothesis-testing experiments will be required to understand the impact of specific interface features on people's ability to understand this complexity.

In the following section, we report on some common themes that emerged from our interviews and comments our participants left in our questionnaires.

### *3.3.2 Thumbs up/down controls*

We asked people about their perceptions of the like/dislike and sharing buttons. We wanted to know what meaning people ascribed to these controls. People found the purpose of the "dislike" button intuitive and its functionality useful. It was used in 12 of 19 tasks, sometimes quite heavily:

*The like/dislike is amazing ... you are able to hide documents as well ... that again simplifies stuff ...* [p5]

*Before, you had to read tens of pages on Google scholar... that's very painful. Some of* [it is] *not useful, so you want to delete them...in this case you just see it, if it's not useful, you delete it, and it disappears from your screen. It's very practical. In the end its just information you want.* [p4]

### *3.3.3 Query interface*

While there was considerable use of the query interface (100 unique queries plus 50 revisits), only seven of these unique queries were relevance feedback queries. We asked about the low response rate for this feature, and found a range of responses. Many people had not realized that the feature existed in the interface, or how they could use it.

[The system] *has collaboration, faceted search... similarity search. In many cases, the feature was not very discoverable. It did not really occur to me that I can combine those features to do my task. I think that was the reason I wound up using mostly query search instead of other things* [p2]

*I haven't looked at this specific feature so I wasn't aware when I was searching… I sometimes randomly enter query terms and continue searching. Maybe I wasn't aware of strategies that are better than current ones.* [p7]

*The "combine queries" takes longer time, and I don't know how it combines queries. Does it combine two queries?* [p9]

Others had *a priori* biases or unmet expectations:

*I typically change queries by myself. I don't combine queries because the system behavior is not predictable. Why don't I just combine them myself?* [p3]

*I think the feature is useful if the results are actually related to the results. But I don't actually find that it is always the case. Even for Google the results are usually not that relevant* [p3]

*I wanted to look at the good documents, rather than looking for more similar documents. If I have good documents already, why would I want to look for more similar documents?* [p8]

*I also liked the fact that you combined faceted search with query search so I could filter [at] the same time type a query. That was another fun feature.* [p2]

Overall, it was clear that while some of the query functionality met expectations, the value of other aspects of the design were hard to understand.

### *3.3.4 System performance and stability*

Certain aspects of the system were too slow for interactive use. In particular, running fusion queries took a long time, as participant 9 pointed out. We also uncovered some bugs during the course of the deployment that prevented all participants from using some of the system features.

Another problem that hampered our evaluation was poor data: the snapshot of CiteSeer that we had obtained contained data only through 2007, and many of the URLs were stale. As CiteSeer crawls the web and collects papers for its database, it often encounters multiple versions of the same paper. An automatic process clusters the papers based on extracted metadata, but each version is associated with its own URL. We took the naïve approach of using the first URL as the representative for each cluster. This proved problematic in some instances because some URLs were no longer valid, and some others pointed to PostScript files that a browser could not easily display.

In addition, some of the automatically-extracted metadata were not correct or were missing. Although we could display the text of the document instead of the PDF, the errors and the unreliable metadata did not improve people's perception of the system.

### 3.3.5 *Lessons learned*

Leveraging people's prior experience with thumbs up/thumbs down controls proved successful, but the lesson needed to be applied to other parts of the interface as well. People were confused by our query expansion design because they lacked prior experience with such tools, and our dataset needed improvement: although people were more engaged with the content, they were frustrated by the errors and omissions in it.

## 4. QUERIUM 2.0

We made some radical changes to the user interface to address user experience and performance issues we discovered during the deployment, some of which had never been adequately addressed in earlier designs. These included unifying the query and relevance feedback interface, integrating the summary and search results views, and introducing a number of design elements to make the interface more engaging.

## 4.1 Reinventing the client

Earlier versions of Querium clients were structured around HTML pages constructed by the server based on user requests, with asynchronous calls to load search results. The client did not have an explicit data model itself; it merely reflected the data structures—topic, query, posting, document—that the server loaded from the database. The server was responsible for retrieving data from the database, and for doing all the computations on that data. One particularly expensive computation was the fusion operation that required the server to identify all documents retrieved within a particular topic by at least two queries. This involved a four-way join, an expensive operation the results of which would change with every query and assessment that the user made. Another database-intensive operation was the computation of retrieval histories for each document. The poor responsiveness of the server, driven largely by these computations, was one of the problems many participants commented on.

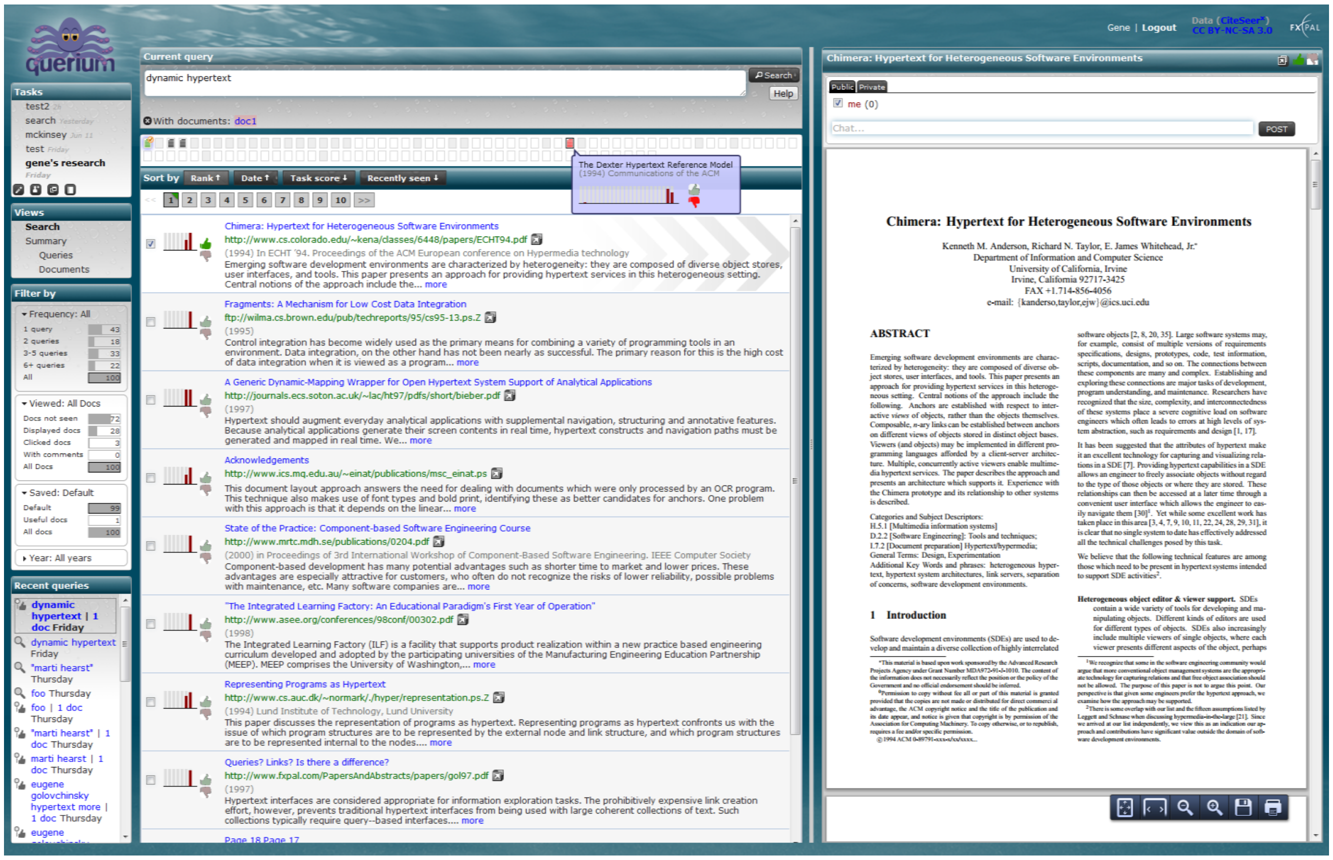
To address this issue, we redistributed the computational burden between the server and the client. The server was simplified to perform three classes of operations: to return the full state of a topic on page load, to run queries, and to manage an event queue.

The client was now responsible for presenting the results, for managing navigation of the search history, for computing the fusion results and the document retrieval history, and for updating the state of the topic based on events from the server. To manage this complexity on the client, we implemented a multi-tier architecture within the client: the client's data graph is kept in sync with the server by periodic updates; views are generated dynamically from this structure.

### 4.1.1 *Client architecture*

One interesting insight from this redesign was that the data model in the client should be different from the model used by the server for persistence. The server made a strong distinction between Documents and Postings: Documents were used to store the metadata shared among all topics, and Postings were used to represent specific retrieval results. The client, on the other hand, aggregated data for a single Topic, making that distinction moot.

The client had to keep track of all the documents retrieved for a given topic, and of the ranks at which they were retrieved. Rather than re-computing this on every request, we aggregate this data immediately upon loading the topic, and then update it incrementally as new events are processed. Our new model handles both the



Gene | Logout Data (CiteSeer^x) CC BY-NC-SA 3.0 FXPAL

**Tasks**
test2 2h
search Yesterday
mckinsey Jun 11
test Friday
**gene's research** Friday

**Views**
**Search**
Summary
Queries
Documents

**Filter by**

Frequency: All

| | |
|---|---|
| 1 query | 43 |
| 2 queries | 18 |
| 3-5 queries | 33 |
| 6+ queries | 22 |
| All | 100 |

Viewed: All Docs

| | |
|---|---|
| Docs not seen | 72 |
| Displayed docs | 28 |
| Clicked docs | 3 |
| With comments | 0 |
| All Docs | 100 |

Saved: Default

| | |
|---|---|
| Default | 99 |
| Useful docs | 1 |
| All docs | 100 |

Year: All years

**Recent queries**
**dynamic hypertext | 1 doc Friday**
dynamic hypertext Friday
"marti hearst" Thursday
foo Thursday
foo | 1 doc Thursday
"marti hearst" | 1 doc Thursday
marti hearst | 1 doc Thursday
eugene golovchinsky hypertext more | 1 doc Thursday
eugene

**Current query**
dynamic hypertext
Search Help
With documents: doc1

Sort by Rank ↑ Date ↑ Task score ↓ Recently seen ↓
<< 1 2 3 4 5 6 7 8 9 10 >>

The Dexter Hypertext Reference Model
(1994) Communications of the ACM

Chimera: Hypertext for Heterogeneous Software Environments
http://www.cs.colorado.edu/~kena/classes/6448/papers/ECHT94.pdf
(1994) In ECHT '94. Proceedings of the ACM European conference on Hypermedia technology
Emerging software development environments are characterized by heterogeneity: they are composed of diverse object stores, user interfaces, and tools. This paper presents an approach for providing hypertext services in this heterogeneous setting. Central notions of the approach include the... more

Fragments: A Mechanism for Low Cost Data Integration
ftp://wilma.cs.brown.edu/pub/techreports/95/cs95-13.ps.Z
(1995)
Control integration has become widely used as the primary means for combining a variety of programming tools in an environment. Data integration, on the other hand has not been nearly as successful. The primary reason for this is the high cost of data integration when it is viewed as a program... more

A Generic Dynamic-Mapping Wrapper for Open Hypertext System Support of Analytical Applications
http://journals.ecs.soton.ac.uk/~lac/ht97/pdfs/short/bieber.pdf
(1997)
Hypertext should augment everyday analytical applications with supplemental navigation, structuring and annotative features. Because analytical applications generate their screen contents in real time, hypertext constructs and navigation paths must be generated and mapped in real time. We... more

Acknowledgements
http://www.ics.mq.edu.au/~einat/publications/msc_einat.ps
This document layout approach answers the need for dealing with documents which were only processed by an OCR program. This technique also makes use of font types and bold print, identifying these as better candidates for anchors. One problem with this approach is that it depends on the linear... more

State of the Practice: Component-based Software Engineering Course
http://www.mrtc.mdh.se/publications/0204.pdf
(2000) in Proceedings of 3rd International Workshop of Component-Based Software Engineering. IEEE Computer Society
Component-based development has many potential advantages such as shorter time to market and lower prices. These advantages are especially attractive for customers, who often do not recognize the risks of lower reliability, possible problems with maintenance, etc. Many software companies are... more

"The Integrated Learning Factory: An Educational Paradigm's First Year of Operation"
http://www.asee.org/conferences/98conf/00302.pdf
(1998)
The Integrated Learning Factory (ILF) is a facility that supports product realization within a new practice based engineering curriculum developed and adopted by the participating universities of the Manufacturing Engineering Education Partnership (MEEP). MEEP comprises the University of Washington,... more

Representing Programs as Hypertext
http://www.cs.auc.dk/~normark/./hyper/representation.ps.Z
(1994) Lund Institute of Technology, Lund University
This paper discusses the representation of programs as hypertext. Representing programs as hypertext confronts us with the issue of which program structures are to be represented by the external node and link structure, and which program structures are to be represented internal to the nodes.... more

Queries? Links? Is there a difference?
http://www.fxpal.com/PapersAndAbstracts/papers/gol97.pdf
(1997)
Hypertext interfaces are considered appropriate for information exploration tasks. The prohibitively expensive link creation effort, however, prevents traditional hypertext interfaces from being used with large coherent collections of text. Such collections typically require query--based interfaces.... more

Page 18 Page 17

**Chimera: Hypertext for Heterogeneous Software Environments**
Public Private
me (0)
Chat... POST

**Chimera: Hypertext for Heterogeneous Software Environments**

Kenneth M. Anderson, Richard N. Taylor, E. James Whitehead, Jr.*
Department of Information and Computer Science
University of California, Irvine
Irvine, California 92717-3425
FAX +1.714-856-4056
e-mail: {kanderso,taylor,ejw}@ics.uci.edu

**ABSTRACT**

Emerging software development environments are characterized by heterogeneity: they are composed of diverse object stores, user interfaces, and tools. This paper presents an approach for providing hypertext services in this heterogeneous setting. Central notions of the approach include the following. Anchors are established with respect to interactive *views* of objects, rather than the objects themselves. Composable, *n*-ary links can be established between anchors on different views of objects stored in distinct object bases. Viewers (and objects) may be implemented in different programming languages afforded by a client-server architecture. Multiple, concurrently active viewers enable multimedia hypertext services. The paper describes the approach and presents an architecture which supports it. Experience with the Chimera prototype and its relationship to other systems is described.

Categories and Subject Descriptors:
H.5.1 [Multimedia information systems]
D.2.2 [Software Engineering]: Tools and techniques;
I.7.2 [Document preparation] Hypertext/hypermedia;
General Terms: Design, Experimentation
Additional Key Words and phrases: heterogeneous hypertext, hypertext system architectures, link servers, separation of concerns, software development environments.

**1 Introduction**

Software development environments (SDEs) are used to develop and maintain a diverse collection of highly interrelated software objects [2, 8, 20, 35]. Large software systems may, for example, consist of multiple versions of requirements specifications, designs, prototypes, code, test information, scripts, documentation, and so on. The connections between these components are many and complex. Establishing and exploring these connections are major tasks of development, program understanding, and maintenance. Researchers have recognized that the size, complexity, and interconnectedness of these systems place a severe cognitive load on software engineers which often leads to errors at high levels of system abstraction, such as requirements and design [1, 17].

It has been suggested that the attributes of hypertext make it an excellent technology for capturing and visualizing relations in a SDE [7]. Providing hypertext capabilities in a SDE allows an engineer to freely associate objects without regard to the type of those objects or where they are stored. These relationships can then be accessed at a later time through a convenient user interface which allows the engineer to easily navigate them [30][1]. Yet while some excellent work has taken place in this area [3, 4, 7, 9, 10, 11, 22, 24, 28, 29, 31], it is clear that no single system to date has effectively addressed all the technical challenges posed by this task.

We believe that the following technical features are among those which need to be present in hypertext systems intended to support SDE activities[2].

**Heterogeneous object editor & viewer support.** SDEs contain a wide variety of tools for developing and manipulating objects. Different kinds of editors are used for different types of objects. SDEs also increasingly include multiple viewers of single objects, where each viewer presents different aspects of the object, perhaps

*This material is based upon work sponsored by the Advanced Research Projects Agency under Grant Number MDA972-91-J-1010. The content of the information does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.

[0]Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commerci al advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copyright is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
©1994 ACM 0-89791-xxx-x/xx/xxxx...

[1]We recognize that some in the software engineering community would argue that more conventional object management systems are the appropriate technology for capturing relations and that free object association should not be allowed. The purpose of this paper is not to argue this point. Our perspective is that given some engineers prefer the hypertext approach, we examine how the approach may be supported.

[2]There is some overlap with our list and the fifteen assumptions listed by Leggett and Schnase when discussing hypermedia-in-the-large [21]. Since we arrived at our list independently, we view this as an indication our approach and contributions have significant value outside the domain of software development environments.

**Figure 9. Querium 2.0 prototype, showing the sidebar with view and filter selections and some recent queries on the left; the query area and search results in the middle, and a selected document on the right.**

document's retrieval history and the fusion operation as follows.

For each document retrieved by a query:

1. For the first instance of a document, create a document object with its metadata, and store it in the topic.
2. Create a new posting for the document, and add it to the document's history, which is a time-ordered list of postings.
3. Add the posting to the query's postings list as well.
4. When a document is assessed, store the assessment in the document, not on the posting (as on the server).

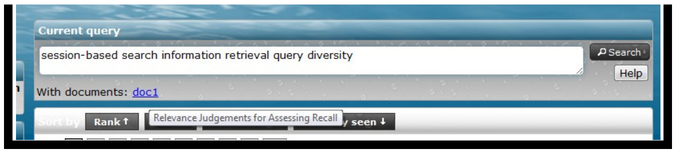
Thus a document's retrieval history is directly available whenever that document is displayed. The fusion view is now simply a list of all documents retrieved in a topic. The fusion view can filter the documents based on the size of each document's history and on whether the documents were viewed, liked, etc.

Since the query results list and the fusion list are displayed almost identically in the interface, it's useful to minimize the differences between documents and postings. We do this by deriving (using JavaScript's prototype inheritance mechanism) the Posting instance from the Document *instance* that represents the retrieved document. Thus while the server models the document-posting relationship as a *has-a* relationship, the client uses *is-a*. This allows the client to treat documents and postings interchangeably, simplifying the query and fusion results display code. Assessments are collapsed into the Document as well, further reducing schema complexity.

We introduced an asynchronous event queue to generate updates to the client. Event notifications are used to send to the client all search results and all data generated by a searcher's collaborators, including chat comments, queries, assessments, etc. The server maintains an event queue for each topic, and places data on the queue based on some user's actions. Data stays on the queue for some time (usually a couple of minutes), giving all clients a chance to poll for it. A new client that joins the session gets a fresh snapshot of the data, which is then updated incrementally.

### 4.1.2 Unified query interface

Querium 1.0 implemented four different kinds of queries—keyword, *ad hoc* relevance feedback (RF), "liked" RF, and fusion—which caused some confusion among our users. To simplify the interface without sacrificing functionality, we refashioned fusion as the document history, dropped the redundant "liked" RF query in favor of *ad hoc* document selection, and integrated the RF mechanism into the keyword query area. (For example, the 'Current query' component in Figure 11 shows that the last query viewed was a relevance feedback query using the first-ranked document; hovering over doc1 shows the title; clicking on that link opens the document in the right-hand document pane.) Instead of four different query functions, we now have one traditional keyword area that also reflects the current RF document selection (Figure 10).



session-based search information retrieval query diversity

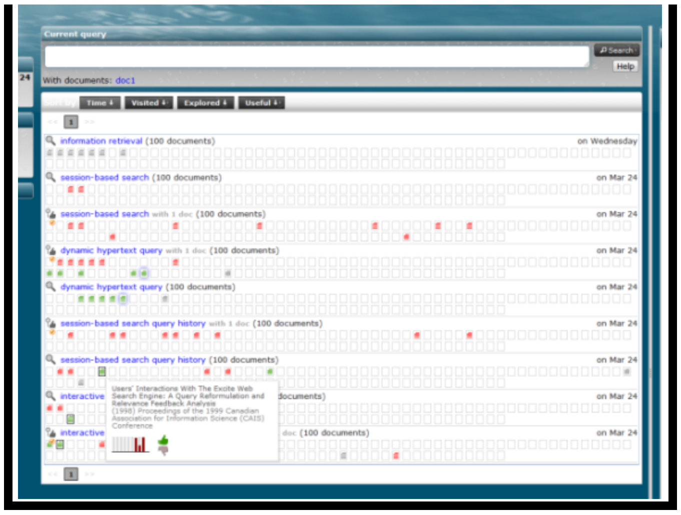
With documents: doc1

Relevance Judgements for Assessing Recall

**Figure 10. Close-up of query box showing a relevance-feedback document; tooltip shows document title.**

### 4.1.3 Integrated summary view

Querium 1.0 introduced a summary view that offered a chronological overview of the search session with some filtering capabilities to show only documents, queries, saved documents, etc. While it provided some potentially useful functionality, the information it displayed was somewhat disconnected from the rest of the interface. Furthermore, query history information was also shown in a drop-down list associated with the query text box (Figure 2). In the redesign, we combined the summary view, the query history and fusion functions into a unified display (Figure 11), and created an integrated view selection mechanism in the sidebar using which a searcher could move between the current results, the query history, or the document history (fusion) views.



With documents: doc1

information retrieval (100 documents) on Wednesday
session-based search (100 documents) on Mar 24
session-based search with 1 doc (100 documents) on Mar 24
dynamic hypertext query with 1 doc (100 documents) on Mar 24
dynamic hypertext query (100 documents) on Mar 24
session-based search query history with 1 doc (100 documents) on Mar 24
session-based search query history (100 documents) on Mar 24
interactive documents) on Mar 24
interactive doc (100 documents) on Mar 24

Users' Interactions With The Excite Web Search Engine: A Query Reformulation and Relevance Feedback Analysis (1998) Proceedings of the 1999 Canadian Association for Information Science (CAIS) Conference

**Figure 11. Query history view (partial screenshot)**

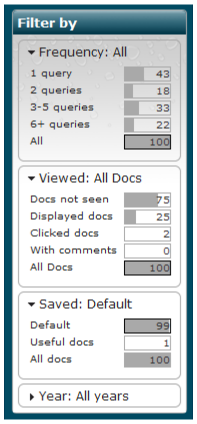
The query history view shows the list of queries, with surrogates for retrieved documents, and annotates each document to indicate whether the document was judged to be useful (green), not useful (red), seen (grey), used for relevance feedback (yellow checkmark), or currently shown in the document pane on the right (halo). When the user points the mouse at any document surrogate, the system highlights all other places in the current view where the same document was retrieved. In addition, a tool-tip popup displays the title, retrieval history, and assessment controls for the document. This allows the searcher to get a quick overview of the results and explore the results without leaving the overview.

We also made it simple to switch among different search tasks: the list of a person's tasks is always shown in the top-left corner, just below the Querium logo. This list is sorted by time, highlighting the few most recently-accessed tasks. If the number of tasks increases beyond four or five, the list becomes scrollable. Below the task list is a set of controls for creating and editing a task, for inviting collaborators, and for opening the task notebook.

### 4.1.4 Filters

The filter interface and data structures were redesigned to provide more control in a consistent manner. While the document metadata remained unchanged, the new design separates process metadata into three categories: retrieval frequency, viewed status, and assessment. Each facet is independent of the others, making it possible to select documents through arbitrary combinations of these facets rather than through the predefined combinations available in the earlier design.

Figure 12 shows three opened process metadata filters, indicating counts of documents in each category. Forty two documents have been retrieved only by one (the current) query, 18 documents have been retrieved by two queries, etc. Of the 100 documents retrieved by this query, 75 have not been seen, and two have been clicked on to view the PDFs. One document has been assessed as useful. Figure 13 shows the same data filtered to show only the newly-retrieved documents. The total count goes down to 42, of which nine snippets have been displayed and 34 remain unexamined.

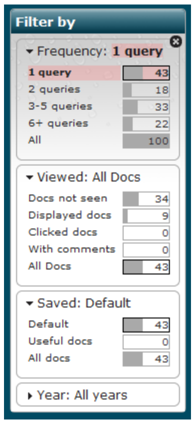


| Frequency: All | |
|---|---|
| 1 query | 43 |
| 2 queries | 18 |
| 3-5 queries | 33 |
| 6+ queries | 22 |
| All | 100 |

| Viewed: All Docs | |
|---|---|
| Docs not seen | 75 |
| Displayed docs | 25 |
| Clicked docs | 2 |
| With comments | 0 |
| All Docs | 100 |

| Saved: Default | |
|---|---|
| Default | 99 |
| Useful docs | 1 |
| All docs | 100 |

Figure 12. Filters showing all documents.



| Frequency: 1 query | |
|---|---|
| 1 query | 43 |
| 2 queries | 18 |
| 3-5 queries | 33 |
| 6+ queries | 22 |
| All | 100 |

| Viewed: All Docs | |
|---|---|
| Docs not seen | 34 |
| Displayed docs | 9 |
| Clicked docs | 0 |
| With comments | 0 |
| All Docs | 43 |

| Saved: Default | |
|---|---|
| Default | 43 |
| Useful docs | 0 |
| All docs | 43 |

Figure 13. Filter showing just newly-retrieved documents.

The display of retrieved documents, which is paginated, has been augmented with a summary of the entire results set above it, as shown in Figure 14. As in the query history view (Figure 11), each rectangle represents a document; and its status—viewed, useful, not useful, etc.—is indicated through color coding. The state of documents retrieved by earlier queries is indicated in this results display directly. For example, the red icon in the middle of the top row was retrieved by an earlier query and assessed as not useful. Even though such documents are suppressed from the detailed results display by default, the summary reminds searchers that such documents have in fact been retrieved. This can help people gauge the quality of the query and makes it easier to revisit earlier decisions in light of new evidence.

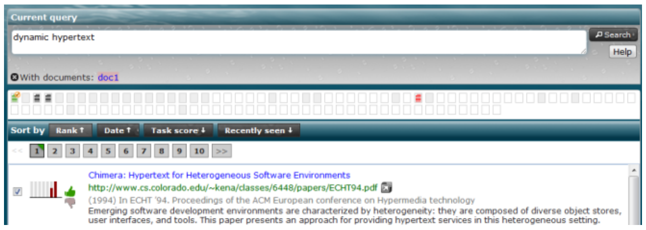
Filtering is also carried over into the results display. The summary indicates which documents have been included and which have been left out by the filters (Figure 15). This display gives a direct impression of how many documents are selected by the filter, and also indicates their distribution in the ranked list.

## 4.2 Engagement and Persuasion

Exploratory search interfaces have to support a more complex set of interactions compared with the familiar web-based search engine interfaces. One challenge is to design useful interactions for exploration; another is to get people to see the value in using such tools and to get them to internalize new ways of thinking about information seeking interfaces.

We saw some of these adoption issues in our evaluations of Querium 1.0, and decided to explore a new approach to improve people's impressions of the system. We are introducing new experimental interactive elements to the Querium 2.0 design that we hope will increase people's engagement with the system. While people's motivation to search for information is typically rooted in some external tasks (e.g., writing a paper, reviewing case law, etc.), their willingness to continue using a tool can be affected by characteristics of that tool [4][9][29].

We have taken a two-pronged approach to exploring this space, tackling engagement and persuasion separately. Engagement here represents tactics to make the use of the system pleasant and responsive [24]; persuasion represents a strategic effort on our part to nudge user behavior into more productive directions [15].

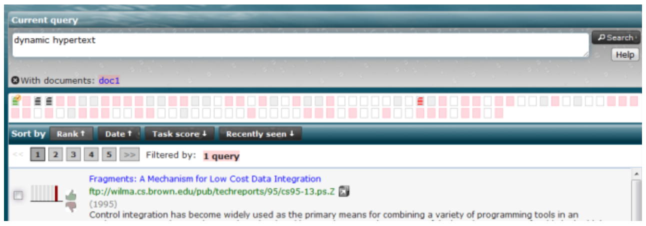


dynamic hypertext

With documents: doc1

Chimera: Hypertext for Heterogeneous Software Environments
http://www.cs.colorado.edu/~kena/classes/6448/papers/ECHT94.pdf
(1994) In ECHT '94. Proceedings of the ACM European conference on Hypermedia technology
Emerging software development environments are characterized by heterogeneity: they are composed of diverse object stores, user interfaces, and tools. This paper presents an approach for providing hypertext services in this heterogeneous setting.

Figure 14. Document results including the overview of the entire result set. Green icons represent useful (bookmarked) documents; grey icons represent clicked-on documents; red icons represent documents marked as not useful.



dynamic hypertext

With documents: doc1

Filtered by: 1 query

Fragments: A Mechanism for Low Cost Data Integration
ftp://wilma.cs.brown.edu/pub/techreports/95/cs95-13.ps.Z
(1995)
Control integration has become widely used as the primary means for combining a variety of programming tools in an

Figure 15. Same results, with a filter applied. Pink rectangles represent documents selected by the filter.

### 4.2.1 *Engagement*

To improve engagement, we restructured views to never scroll off the screen, added some transitional animations to reinforce people's actions, and introduced a colorful background image against which application components are positioned.

The various views are now kept from extending beyond the size of the window, making the interface feel more like an application rather than a web page. Where scrolling may be necessary (such as in the display of search results), it is kept to a minimum by paginating the list and only scrolling a small area of the view.

Earlier designs had one key animation: when a document was marked as "not useful" or "disliked," it was faded out prior to hiding it from the default view. This was well-received in our Querium 1.0 evaluation: people found the action comprehensible and visually pleasing, and, as a result, were willing to use the control. We built on it by introducing a sliding arrow animation (shown in Figure 9) that shows which document from the results list is currently displayed. The status bar showing informational and error messages is now animated: it drops down from the top of the screen, and then retracts after a few seconds.

### 4.2.2 *Persuasion*

We have also started investigating persuasive aspects of the interface. The goal is to gradually reinforce searchers' behaviors that lead to productive outcomes over those that don't. The design space here is potentially rich and quite under-investigated, both in terms of which behaviors to reward and also in terms of the kinds of rewards. Recent work in this space has taken an explicit approach to educating users by offering timely tooltips [23] and by allowing people to compare their behavior to that of "experts" [7]. Our design relies more on nudging people toward more effective behaviors through ambient displays.

Exploratory search in digital libraries can benefit from a number of behaviors, including running diverse queries [14] and making relevance judgments [10]. We would like to get people to run multiple, diverse queries because that is an effective way of understanding the breadth of a topic. We would also like people to make many judgments of usefulness, as that helps us do relevance or pseudo-relevance feedback and gives us a sense people's progress (or lack thereof) toward their goals. Unfortunately, people may not be aware of the unity of these tactics.

We are experimenting with two methods of implementing persuasion, one for reflecting effects of a user's actions (aggregate or incremental), and one for encouraging exploration of the interface when a particular feature has not been used.

### 4.2.3 *Reflecting desirable outcomes*

Our generalized approach to reflecting effects of prior actions is to identify a metric of behavior or of system performance that correlates with desired outcomes, and then to render that metric in some manner in the interface. Possible metrics include the number of relevance judgments, the number of unique documents retrieved in a task, the number or DCG score of new or unique documents retrieved in a query, etc. These metrics can then be rendered in the user interface in some manner.

Our initial implementation maps query diversity onto the "murkiness" of the background graphic. Diversity is computed as the fraction of the query's results that were retrieved for the first time by that query. (Subsequent queries can retrieve some of the same documents without affecting the scores of preceding queries.) The murkiness effect is achieved by superimposing two similar images, a darker one on top of a lighter version, and then varying the opacity of the darker image to correspond to the computed diversity score. When an earlier query is selected, or a new query is run, the opacity change is animated over a few seconds to create an ambient display that does not interrupt reading or text-related sense-making activity of the searcher [1].

### 4.2.4 *Encouraging desirable behavior*

To encourage people to experiment with the interface, we are developing a user model that records people's behavior and system performance, and suggests behavioral tactics that may be more effective. For example, if a user has only run keyword queries for a while, and the number of newly-retrieved documents is low (that is, the same documents are getting re-retrieved by consecutive queries), the model suggests that relevance feedback be tried to explore new parts of the collection. In our preliminary implementation, this causes the relevance feedback controls to be highlighted subtly. The intent is to draw the searcher's attention to these controls without interrupting current activity.

Our next steps in this area are to investigate more systematically how best to structure user behavior models, and what sorts of visual or other displays are appropriate for different kinds of feedback. These kinds of user models can be based on simple *ad hoc* thresholds that seem to produce sensible results, or they can be quite complex and dynamic, incorporating user-specific preferences and behavioral patterns. One interesting experiment to consider is how a simple model based on thresholds compares to more complex algorithms.

The second aspect that bears investigation is what sorts of visual effects are effective at encouraging positive behavior. In addition to varying background or other decoration color, hues, saturation, opacity, etc. it may be possible to introduce animations that are consistent with the metaphor the interface. In our case, the aquatic theme may be exploited to introduce animated sprites of fish and other marine life that may occasionally appear in the interface. We can tie the frequency of appearance of these (hopefully!) entertaining animations to the saving of useful documents, to instances of effective collaboration, etc. A design challenge is to balance the disruption of these explicit animations with the surprise and delight that they may evoke.

The exact space of actions and schedule of rewards needs to be discovered empirically, but the idea that playful behavior for a serious purpose can improve people's attitudes toward a system and can be used to foster desirable behaviors seems interesting and promising to us.

## 5. CONCLUSION

Golovchinsky and Pickens [17] proposed a model of propagating context among information retrieval objects in the course of an interactive session. Our design process started with the naïve implementation of several such context flows, including several ways to select documents for relevance feedback, and a mechanism for merging results of multiple queries.

While technically we were able to build several versions of such context-preserving interactions, we found that people did not understand how to use some of them. The lack of positive response was due in part to some usability problems with some of the designs, but also in part due to expectations people had of search tools, expectations developed through exposure to modern web search interfaces.

Does Google make us stupid? [12] [3] Not exactly. But it does reward certain behaviors, borne of their design goals to minimize the computational effort required to deliver search results, and to make interfaces accessible to most people with the least amount of effort. In trying to support more complex information seeking behaviors, we are therefore faced with users' expectations established over repeated encounters with web search engines. Thus the work presented here can be seen as an ongoing attempt to strike a balance between functionality and simplicity, between meeting users' expectations and nudging them toward more appropriate interfaces for these complex tasks.

Through our iterative design process, we considered a range of factors from server performance to the user experience, and tried to make design decisions that empower searchers. We adopted a light-weight, informal evaluation strategy to capture the important aspects of our designs. None of our evaluations is complete or definitive; we have aimed for the light-weight, the indicative, the opportunistic. For this stage in our design process, we prefer heuristic evaluations, design reviews and interviews to more formal, large-scale evaluations. This iterative design yielded some insights that we believe generalize beyond the current system.

Leveraging the familiar need not be a slavish imitation of existing tools or interfaces: we capitalized on people's familiarity with the thumbs-up/thumbs-down interface to implement a relevance feedback interface that was comprehensible and yet introduced new functionality. The design challenge is to identify such constructs that can be embraced and subverted.

Interactivity and responsiveness matter, and systems that favor functionality over responsiveness do so at the peril of poor adoption. Designers of these kinds of systems must consider the whole system and the interplay among its components, rather than designing the interface and the back end independently. It took us a while to recognize the inertia of the original design had to be overcome with a radical refactoring of the system to generate a more responsive, and therefore more usable, interface. The lesson here is: refactor early, refactor often.

Finally, in the process of dealing with the challenges of creating a usable interface for a complex task, we introduced the notion that ambient displays and persuasive techniques should be applied to the design of interfaces for information seeking. We argue that engaging interfaces should improve people's satisfaction and may encourage more interaction with and exploration of the tools themselves. We expect that this increased engagement will lead to more learning and will increase users' search expertise over time. We expect to test these hypotheses in the coming months when Querium is deployed publicly.

## 6. ACKNOWLEDGMENTS

We thank Larry Rowe and Lynn Wilcox for supporting this research, Stephen Carman for facilitating access to the CiteSeer collection, Andreas Girgensohn for some timely debugging help, and Jeremy Pickens, Diane Kelly, Pernilla Qvarford and David Harper for some early feedback.

## 7. REFERENCES

[1] Adamczyk, P.D. and Bailey, B.P. (2004) If not now, when?: the effects of interruption at different moments within task execution. In *Proc. CHI '04*. ACM Press, pp. 271-278.

[2] Ahn, J.-W., Brusilovsky, P., and Farzan, R. (2005). Investigating users' needs and behavior for social search. In Proc. of the Workshop on New Technologies for Personalized Information Access (held in conjunction with UM'05), Edinburgh, Scotland, UK; pp. 1-12.

[3] Anderson, J. Q. (2010) The Future of the Internet. *Pew Research Center's Internet & American Life Project*. Available online at http://pewinternet.org/topics/Future-of-the-internet.aspx

[4] Attfield, S., Kazai, G., Lalmas, M., Piwowarski, B. (2011) Towards a Science of User Engagements. In Proc. *Workshop on User Modeling for Web Applications*, held in conjunction with *WSDM 2011*, Feb 9-12, 2011.

[5] Aula, A., Jhaveri, N., and Käki, M. (2005) Information Search and Re-access Strategies of Experienced Web Users. In *Proc. WWW 2005*.

[6] Bates, M. (1989) The Design of Browsing and Berrypicking Techniques for the Online Search Interface. *Online Review*, vol. 13 no. 5 pp. 407-24.

[7] Bateman, S., Teevan, J., and White, R. (2012) The search dashboard: how reflection and comparison impact search behavior. In *Proc CHI 2012*, Austin, TX, ACM Press, pp. 1785-1794.

[8] Bharat, K. (2000) SearchPad: Explicit Capture of Search Context to Support Web Search. In *Proc. WWW2000*, pp. 493-501.

[9] Bickmore, T. (2003) *Relational Agents: Effecting Change through Human–Computer Relationships*. Ph.D. thesis, MIT Media Arts and Science.

[10] Buckley, C., Salton, G., and Allan, J. (1994) The effect of adding relevance information in a relevance feedback environment. In *Proc. SIGIR '94*, Springer-Verlag, pp. 292-300.

[11] Capra, R., Marchionini, G., Velasco-Martin, J., and Muller, K. (2010) Tools-at-hand and learning in multi-session, collaborative search. In *Proc. CHI2010*, ACM Press, pp. 951-960.

[12] Carr, N., (2008) Is Google Making us Stupid? *Atlantic Monthly*, July 2008.

[13] CiteSeer. Available online at http://citeseerx.ist.psu.edu/

[14] Evans, B. M. & Chi, E. H. (2010) An elaborated model of social search. *IP&M*, 2010, 46, 656-678

[15] Fogg, B.J. (2002) *Persuasive Technology: Using Computers to Change what We Think and do*. Jonathan Grudin, Jakob Nielsen, and Stuart Card (Eds.). Science&Technology Books.

[16] Golovchinsky, G. (1997) Queries? Links? Is there a difference? In Proc. CHI 1997. ACM Press.

[17] Golovchinsky, G., and Pickens, J. (2010) Interactive Information Seeking via Selective Application of Contextual Knowledge. In *Proc. IIiX 2010*. ACM Press.

[18] Hoppe, H.U. and Zhao, J. (1994). C-TORI: an interface for cooperative database retrieval. In Karagiannis, D. (Ed.), *Database and Expert Systems Applications*, (pp. 103-113). Berlin: Springer-Verlag.

[19] Ingwersen, P. and Järvelin, K. (2005) *The turn: integration of information seeking and retrieval in context.* Springer.

[20] Jhaveri, N. & Räihä, K.-J. (2005) The advantages of a cross-session web workspace, In *CHI Extended Abstracts*, ACM Press, pp. 1949-1952.

[21] MacKay B., Watters C. (2008) Exploring multi-session web tasks. In *Proc. CHI 2008*, ACM Press, pp. 1187-1196.

[22] Marchionini, G. (2006) Exploratory search: from finding to understanding. *CACM*, 49 (4), pp. 41-46.

[23] Moraveji, N., Russell, D., Bien, J., and Mease, D. (2011) Measuring improvement in user search performance resulting from optimal search tips. In *Proc SIGIR 2011*, Beijing, China, ACM Press, pp. 355-364.

[24] Moridis, C., and Economides, A. A. (2008) Towards computer-aided affective learning systems: a literature review. *Journal of Educational Computing Research,* 39 (4), pp. 313-337. Baywood Publishing Co.

[25] Morris, M.R. (2008) A Survey of Collaborative Web Search Practices. In *Proc. CHI 2008*, ACM Press pp. 1657-1660.

[26] Morris, D., Morris, M.R., and Venolia, G. (2008) SearchBar: A Search-Centric Web History for Task Resumption and Information Re-finding. In *Proc. CHI 2008*, 1207-1216.

[27] Pickens, J., Cooper, M., and Golovchinsky, G. (2010) Reverted indexing for feedback and expansion. In *Proc. CIKM 2010*, ACM Press.

[28] Sellen, A., Murphy, R., and Shaw, K. How Knowledge Workers Use the Web. In *Proc. CHI 2002*, ACM Press, pp. 227-234.

[29] Tractinsky, N, Katz, A.S., and. Ikar. D., (2000) What is beautiful is usable. *Interacting with Computers* (13):127-145.

[30] Twidale, M. and Nichols, D. M. (1998) Designing interfaces to support collaboration in information retrieval. *Interacting with Computers* 10(2), pp. 177-193.

[31] White, R. and Roth, R. (2009) Exploratory Search: Beyond the Query-Response Paradigm. Morgan and Claypool.