Exploring and investigating: Supporting high-level search activities

Gina Venolia
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
gina.venolia@microsoft.com

Meredith Ringel Morris
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
merrie@microsoft.com

Dan Morris
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
dan@microsoft.com

Microsoft Research Technical Report MSR-TR-2007-05

Abstract
Much work has been done to improve search as an isolated act, yet little has been done to understand search as it relates to higher-level patterns of behavior, or to develop user interfaces to support these patterns. In this paper, we analyze exploratory and investigative search processes that involve performing several related searches over a period of time. We then discuss requirements for interfaces to support these tasks, and describe three prototype systems.

Keywords
Exploratory search, sensemaking, persistent search.

ACM Classification Keywords
H.5.2 Information Interfaces and Representation: User Interfaces.

Introduction
Broder presents a taxonomy of web search tasks [2], characterizing them as *navigational*, *informational*, or *transactional*. For example a user may want to find the web page for the Toyota Prius car (navigational), find the gas mileage rating of that car (informational), or use a web-based tool to configure the options on the car and estimate costs (transactional). This model presents each query as an isolated task.

Various researchers have identified and described larger-scale patterns of search-related behavior. We have identified two such patterns, which we call *exploration* and *investigation*. In exploration, the user's goal is to increase his or her knowledge about a topic. In investigation, the goal is to reach a decision. In this paper we briefly survey related work, describe these patterns, and then briefly describe three UI prototypes which support them.

Related work

Several researchers have identified large-scale information-seeking patterns that involve search. Vélez et al. [4] present a model of *query refinement* in which a search task consists of a sequence of related queries, where the user incrementally modifies the query specification to achieve a desirable result set.

Bharat [1] describes a pattern of information-seeking behavior, which he identifies with these characteristics: The user searches on many unrelated topics in parallel, often with many browser windows; a given search for information may extend over many sessions; for each information need the user uses many queries, often by a process of query refinement; the user may try the same query on many search services; some users may look at more than one search result page; and when the user finds a useful result, he is often unsure whether it is the best available or if further search is necessary. Bharat describes a tool, *SearchPad*, which attempts to support this pattern of behavior. The tool records the sequence of queries and allows the user to flag important results as *leads*. The SearchPad UI shows the recent queries and, for each, its leads. The tool makes it easy to issue the same query to multiple search engines. Bharat describes how the lightweight and transient nature of this information makes it better suited to support complex searches than bookmarks.

Russell et al. define *sensemaking* [3] as "the process of searching for a representation and encoding data in that representation to answer task-specific questions." They describe a system of three related *learning loops*, wherein representations are created, instantiated, and modified. Sensemaking activities may incorporate multiple actors and take place over long periods of time.

Case study: Root-cause analysis of software defects

One task that relies heavily on in-depth exploration is *root-cause analysis*, or *RCA*, which is the process of finding the reasons for critical failures. We have interviewed one of the people at a large software corporation responsible for RCA of key software defects.

The process begins when the a particular incident is identified as worthy of investigation, e.g. a build break or a critical bug regression introduced by a patch. He chooses some keywords from the incident description and searches for them over the various repositories that he has at his disposal, including the bug database, the product-support knowledgebase, email repositories, project-specific databases, code check-ins, etc. Each repository has its own unique search interface, so this process is quite tedious. He refines his query to improve the result set. He then scans the result list to find potentially-relevant documents, which he opens and examines.

For each relevant document he creates an entry in a structured Microsoft Word document, clearly identifying its date, the document type and name, often a quotation from the document, and often his own notes about the document and/or questions for further investigation. These entries are sorted chronologically, forming a timeline. The Word document is his working "notebook," and is generally not shown to others.

In reading the source documents he may encounter additional terms, people, date ranges, etc. which lead to more searches, which in turn lead to more documents, which then lead to more entries in the timeline. In addition, he is creating and discarding hypotheses, which he retains in another section of his "notebook" document.

In addition to his electronic investigations, he interviews key participants in the incident. These interviews may result in the addition of entries to the timeline and clarifying comments for existing entries.

His final report is a separate document, in which he presents a narrative of the incident and a summary of its root causes. The former is based on the timeline and the latter on the hypotheses.

One of his investigations was of a particular build break. The timeline for this investigation contained about 50 entries representing documents or events that spanned 48 days. The final report identified 21 issues and three root causes. He reported that the investigation, including the electronic sleuthing, interviews, and writing, took him "a couple of weeks" of full-time work.

Explorations and investigations

The pattern of behavior described for RCA is familiar to anyone who has made a purchasing decision or performed a literature review using online resources. Based on our study of RCA, as well as reflection on our own exploratory and investigative search habits, we propose a preliminary model of this pattern.

In the *exploration* pattern of behavior, the user's goal is to increase his or her knowledge about a topic. For example a user might explore the models of environmentally-friendly cars that are available and investigate their environmental benefits compared to standard cars.

An exploration involves finding documents that may be relevant to the topic, examining them, and then discarding irrelevant documents and retaining relevant ones. We call this process *triage*. Documents may be found by executing queries, browsing, or by a combination of the two. While queries may be formed by iterative query refinement, an exploration is often comprised of multiple independent queries. When examining documents a user may find keywords that are then used in further queries.

For simple explorations the relevant documents and keywords may be retained mentally. Complex explorations require external representation, in which the user may record and annotate the relevant documents. A user may invest significant time into an exploration, which may be divided over multiple sessions, so persistence of the accumulated state is important. A user may revisit key searches in an exploration to see "what's new."

The two most important aspects of explorations are that they *persist* and *consist of multiple independent queries*. The multiple independent queries that characterize explorations are distinct from the multiple *related* queries that characterize query refinement.

In an *investigation* the user's goal is to reach a decision. The investigation pattern extends the exploration pattern with a collection of hypotheses, which may be created, supported, and refuted in the course of the exploration. For example a user might perform an investigation to decide which hybrid car models to test-drive. The hypotheses in this case might be "the Prius is the right car for me," "the Civic Hybrid is the right car for me," etc. Each hypothesis is associated with the search results that *support* and *refute* it. An investigation may culminate in the generation of a report describing, at a minimum, the well-supported hypotheses.

Figure 1 shows an idealized schematic representation of the workflow in explorations and investigations. It should be noted that real-world actions (e.g., test-driving cars or getting friends' opinions) fit in with this workflow, but are not depicted. These actions can provide new keywords, documents, notes, and hypotheses.

This model incorporates concepts identified in the Related Work section. The search/refine loop is equivalent to query refinement [4]. Items retained as supporting evidence are similar to the "leads" that SearchPad provides [1]. Sensemaking [3] revolves around representations. In this model the core representation is a collection of documents marked as relevant. Additional aspects of representation come

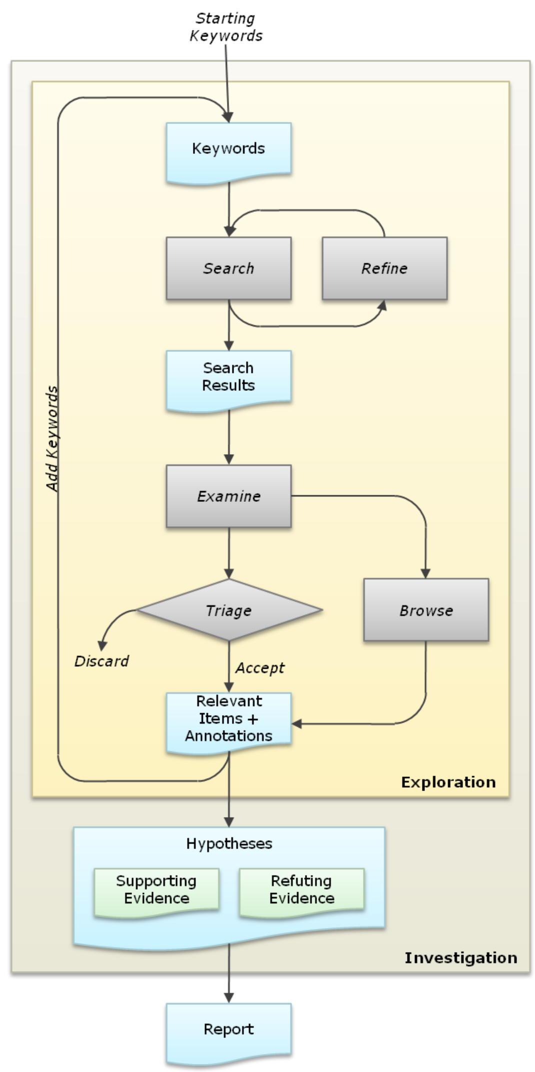

Figure 1: An idealized schematic representation of the workflow in explorations and investigations.

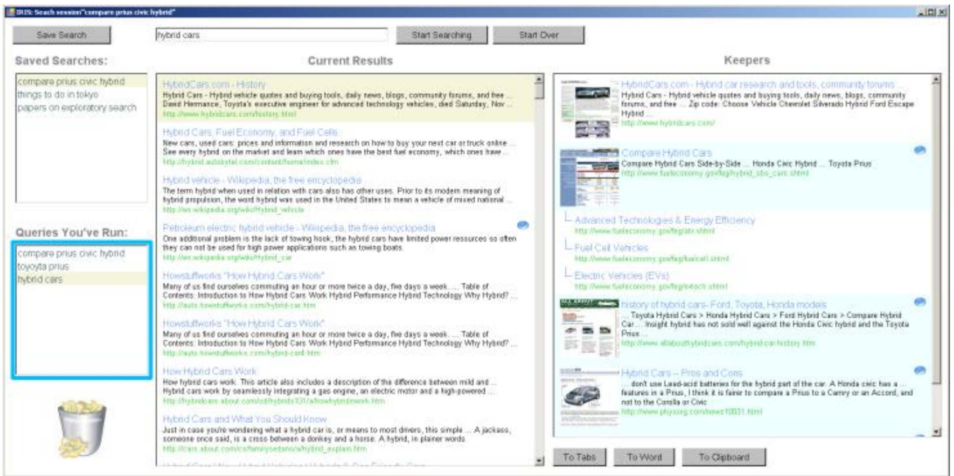

Figure 2: The Interactive, Rapid-Iteration Search (IRIS) prototype.

from annotation and hypotheses. Sensemaking's "residue" becomes additional keywords. The model may be seen as a specific framework for a subset of sensemaking activities.

Not all searches are part of an exploration or investigation. We mean to extend, not replace, Broder's taxonomy [2].

UI to support explorations

The key UI features needed to support exploration are:

- Iterative query refinement
- Juxtaposition of multiple search result sets
- Triage into "evidence" and "discarded" categories
- Manual addition of documents and notes as evidence
- Annotation of evidence with textual comments, tags, or other typed metadata
- Persistence
- Sharing
- Change awareness

We have prototyped three UIs that implement some or all of these features. Each UI emphasizes different aspects of the design space.

The Interactive, Rapid-Iteration Search (IRIS) prototype (Figure 2) emphasizes the following features that we propose will enhance the exploratory search experience:

- *Re-ranking of results to reflect their occurrence among multiple queries*. Results that have been recalled numerous times but ignored can be automatically hidden to remove redundancy, and results that have been recalled by multiple queries but were not highly ranked in any one query are automatically promoted to higher rank.
- *Integration with navigation*. A user can view results in a separate browser without disrupting his search context, and pages that the user subsequently navigates to are automatically associated with the search results that originated the navigation.
- *Automatic summary preparation.* The user can prepare and distribute a summary of his search results, with information about the queries that produced each result, with a single click.

The Persistent Search prototype, shown in Figure 3, emphasizes a document model for searches. A session (i.e., a series of queries related to a single task) can be saved to and loaded from file. This file (which includes

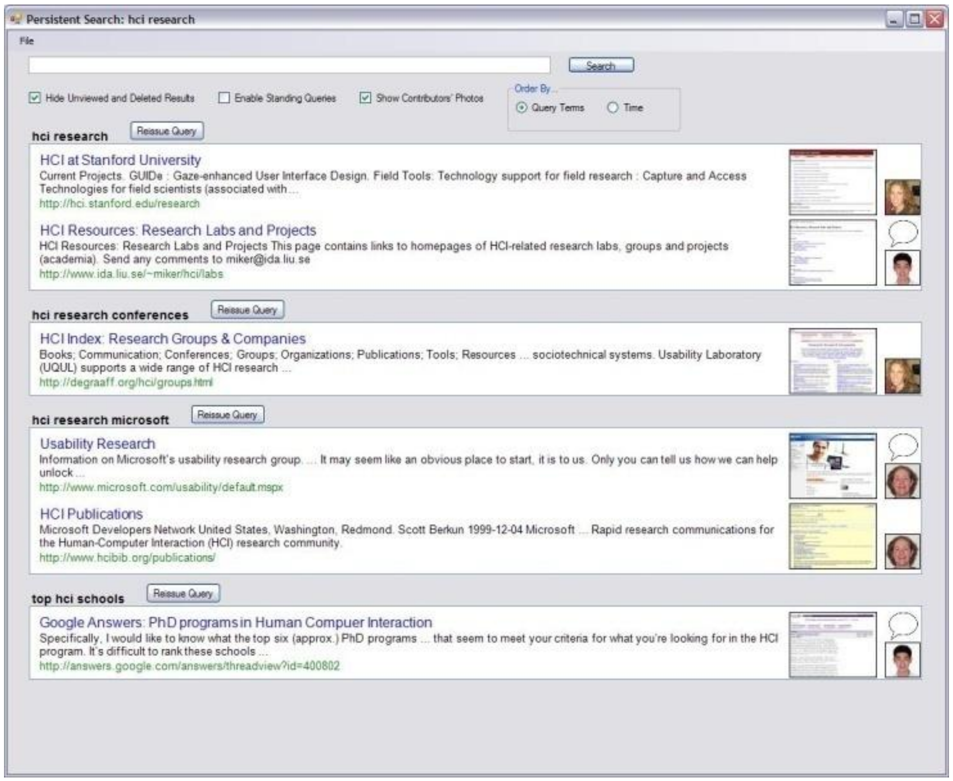

Figure 3: The Persistent Search prototype.

data such as query terms used, sites visited, and comments) can even be shared with others, enabling asynchronous collaboration on an exploratory search task. Each site and comment is tagged with the username of the person who contributed it to the persistent search session.

The Search Grid prototype, shown in Figure 4, also uses a document model. Its user interface is dominated by a grid where the rows are the search results, columns are the queries, and cells show a checkmark for all queries that produced a given result. The list of relevant and untriaged items is the most salient feature; queries are demoted to metadata on the items. Another interesting aspect of the Search Grid prototype is that each item presents commands (in the context menu) to create new queries related to the item, e.g. searching for pages that link to the item's URL or contain text similar to the item's. To some degree these item-relative queries, which we call *pivots*, bridge the gap between searching and browsing.

Extending the UI to support investigations

Any of these prototypes could be extended to support investigations by adding these key UI features:

- Representation of hypotheses
- Association of each hypothesis with supporting and conflicting evidence
- Generation of a skeleton report by emitting a document listing the hypotheses and, for each, the related evidence

For example, in the Search Grid prototype each hypothesis would be represented as a column, containing a three-way toggle so the user could specify whether the item supported or refuted the hypothesis, or was unrelated to it.

Next steps

We are at a preliminary stage in our development of these concepts. There is much to do to validate our model of exploration and investigation and our UI solutions to support those activities. We plan to improve our prototypes and then perform lab studies to

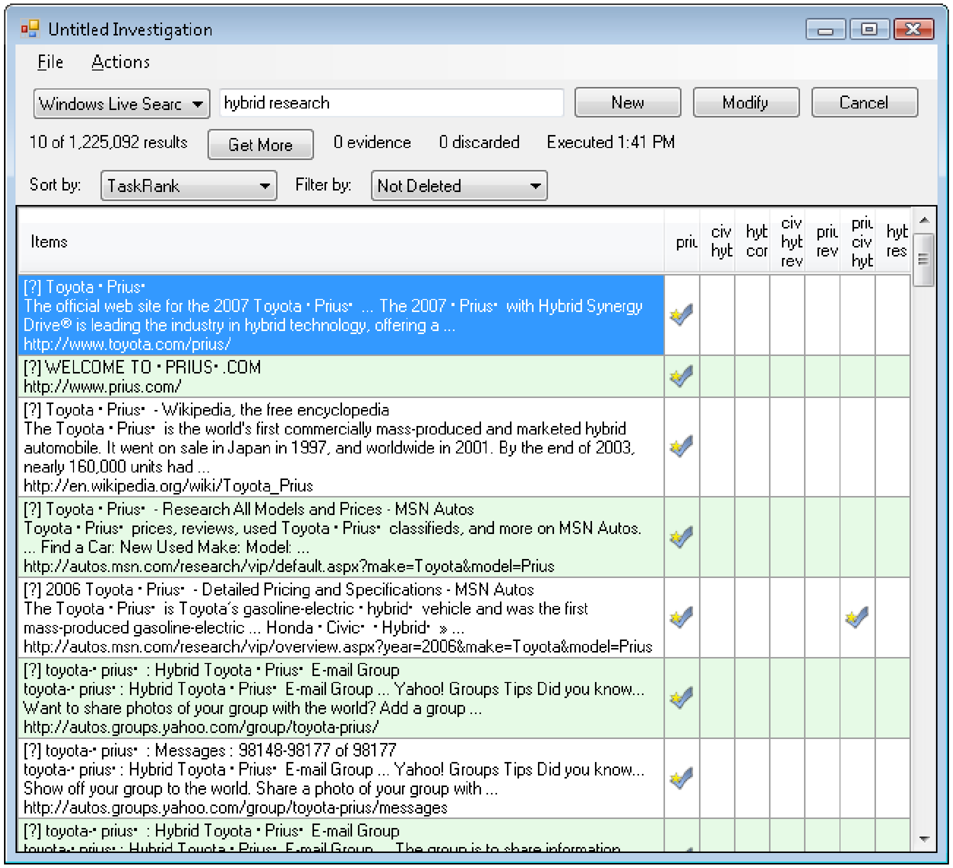

Figure 4: The Search Grid prototype.

see how users approach explorations and investigations with existing tools and our new tools, to detect usability problems in our tools, and to compare performance and user experience to that achieved with traditional search tools. We expect the UI approaches to evolve through this process.

References

[1] Bharat, K. SearchPad: Explicit capture of search context to support Web search. In *Proc. WWW 2000*, 493-501.

[2] Broder, A. A taxonomy of web search. In *SIGIR Forum* 36, 2 (Sep. 2002), 3-10.

[3] Russell, D., Stefik, M., Pirolli, P., and Card, S. The cost structure of sensemaking. In *Proc. CHI 1993*, 269-276.

[4] Vélez, B., Weiss, R., Sheldon, M. A., and Gifford, D. K. Fast and effective query refinement. In *Proc. SIGIR 1997*, 6-15.