

# Facilitating data exploration with query previews: a study of user performance and preference

EGEMEN TANIN<sup>1,2</sup>, AMNON LOTEM<sup>2</sup>, IHAB HADDADIN<sup>3</sup>,  
BEN SHNEIDERMAN<sup>1,2</sup>, CATHERINE PLAISANT<sup>1</sup>,  
LAURA SLAUGHTER<sup>4</sup>

<sup>1</sup>Human-Computer Interaction Laboratory, Institute for Advanced Computer Studies,

<sup>2</sup>Department of Computer Science,

<sup>3</sup>Department of Systems Engineering,

<sup>4</sup>College of Library and Information Services

University of Maryland at College Park, College Park, MD 20742, USA

e-mail: {egemen,lotem}@cs.umd.edu, u-ihaddadin@bss2.umd.edu,

{ben,plaisant}@cs.umd.edu, lauras@wam.umd.edu

**Abstract.** Networked and local data exploration systems that use command languages, menus, or form fillin interfaces rarely give users an indication of the distribution of data. This often leads users to waste time, posing queries that have zero-hit or mega-hit results. Query previews are a novel visual approach for browsing databases. Query previews supply users with data distribution information for selected attributes of the database, and give continuous feedback about the size of the result set as the query is being formed. Subsequent refinements might be necessary to narrow the search. Because there is a risk that query previews are an additional step, leading to a more complex and slow search process, we ran a within subjects empirical study with 12 subjects who used interfaces with and without query previews and with minimized network delays. Even with 12 subjects and minimized network delays we found statistically significant differences showing that query previews could speed up performance 1.6 to 2.1 times and lead to higher user satisfaction.

## 1. Introduction

Exploring information from databases has always been an important research topic in human-computer interaction. Rapid evolution of

networked databases and the amount, type, and format of the data in networked or local databases make problems even more challenging.

Examples of database exploration on the World Wide Web include finding restaurants, homes, employees, jobs, or documents. Users typically enter attribute values to a form fillin interface that has text entry fields (Shneiderman, Byrd, and Croft 1997). This form fillin approach requires the explicit submission of a form to a search engine with three typical results:

- a small set of records that users can look at easily,
- a huge set (mega hit) of related or unrelated records that is burdensome to browse, or
- zero hits.

Users wish for the first result. Nobody wants to browse a huge set of records, only a few of which might be relevant to their needs. Even worse, the case of zero hits lead users to think that they have done something wrong (without any indication of whether a spelling mistake or lack of data is causing the problem). Often, time of the users as well as network and processing resources are wasted.

A common problem is that the interface that is supposed to guide users to a reasonable

result confuses them. The system takes control away from users for long periods of time and does not provide guidance that leads to a successful result. Researchers are exploring improved methods for more efficient browsing of databases. The Rabbit system, by Williams and the work of Heppe, Edmondson, and Spence were early demonstrations of the benefits of progressive querying (Williams 1984, Heppe, Edmondson, and Spence 1985). Other systems show relevance of results and propose a user interface solution, for example Veerasamy and Navathe used histograms, and Hearst used TileBars to visually present relevance of results to the terms used in the query (Veerasamy and Navathe 1995, Hearst 1995). WebTOC uses a hierarchical outliner with bar graph presentations to preview the size and type of items within each branch (Nation, Plaisant, Marchionini, and Komlodi 1997). Eick proposes to augment sliders of visualization systems with density plots or bar charts (Eick 1994).

Dynamic queries use a direct manipulation approach to facilitate query formulation with a visual representation of query components and results (Ahlberg and Shneiderman 1994, Shneiderman 1994, Williamson and Shneiderman 1992). They allow rapid, incremental, and reversible control of the query. Results are presented visually in less than  $1/10^{\text{th}}$  of a second (Shneiderman 1994). Continuous feedback guides users in their query formulation. Figure 1 shows an example dynamic query interface. The application of dynamic querying to general querying environments (i.e. networked) is promising. But high system-resource demands make dynamic querying less applicable to large or networked information collections. Dynamic queries require immediate access to data (i.e. in local memory cells) so that continuous immediate feedback (in less than  $1/10^{\text{th}}$  of a second) is always given to the user. One solution to this problem is to use data aggregation in tandem with dynamic queries (Goldstein and Roth

1994). Another solution might be to use overviews (North, Shneiderman, and Plaisant 1996) and divide the bigger problem into several smaller problems, as in query previews (Doan, Plaisant, and Shneiderman 1996, Plaisant, Bruns, Doan, and Shneiderman 1999). The paradigm of query previews is to give an overview of the database to users before the details are visualized. It divides the querying process into steps to reduce the resource needs for forming the final query. Hence, a smaller and more interesting portion of a larger data set can be downloaded to a local memory cell from the network.

We applied these principles in the development of a two-phase query strategy (preview and refinement) for NASA's Earth Observing System Data Information System (EOSDIS) (Plaisant, Bruns, Doan, and Shneiderman 1999). This strategy is now available as an experimental interface (Greene *et al.* 1999) for the Global Change Master Directory ([gcmd.nasa.gov](http://gcmd.nasa.gov)) and is the basis for the Global Land Cover Facility interfaces ([glcf.umiacs.umd.edu](http://glcf.umiacs.umd.edu)).

### 1.1. *Two-phase query strategy*

For the two-phase approach, the designer has to choose a few discriminating attributes of the database – usually the most commonly used - for the first phase, which is the query preview. The other attributes are kept for the second phase that will include all of the attributes. When the querying environment is activated the query preview appears first. Users make some decisions on this first interface and then move to the second one, the query refinement, to complete the query.

#### *Query preview*

The query preview uses rough values on the data that is being explored. It shows the discriminating attributes of the database so that any selection would lead to a smaller subset of

the database. In order to guide users in the query formulation process, the query preview provides aggregate information about the database. Distribution of data over attribute values is shown graphically (e.g. as a pie or a bar chart). When users select a value on any of the attributes of the query preview panel, the rest of the user interface (e.g. the bars) is updated in less than one second. This is called

tight coupling. Therefore, for each action users take, feedback is given immediately. As users see the potential size of their result set before refining the rough ranges, they are less likely to submit queries that return zero or mega hits. The system load will be reduced if users do not waste their time with zero hit queries or consume network and computing resources in downloading and finding useless results.

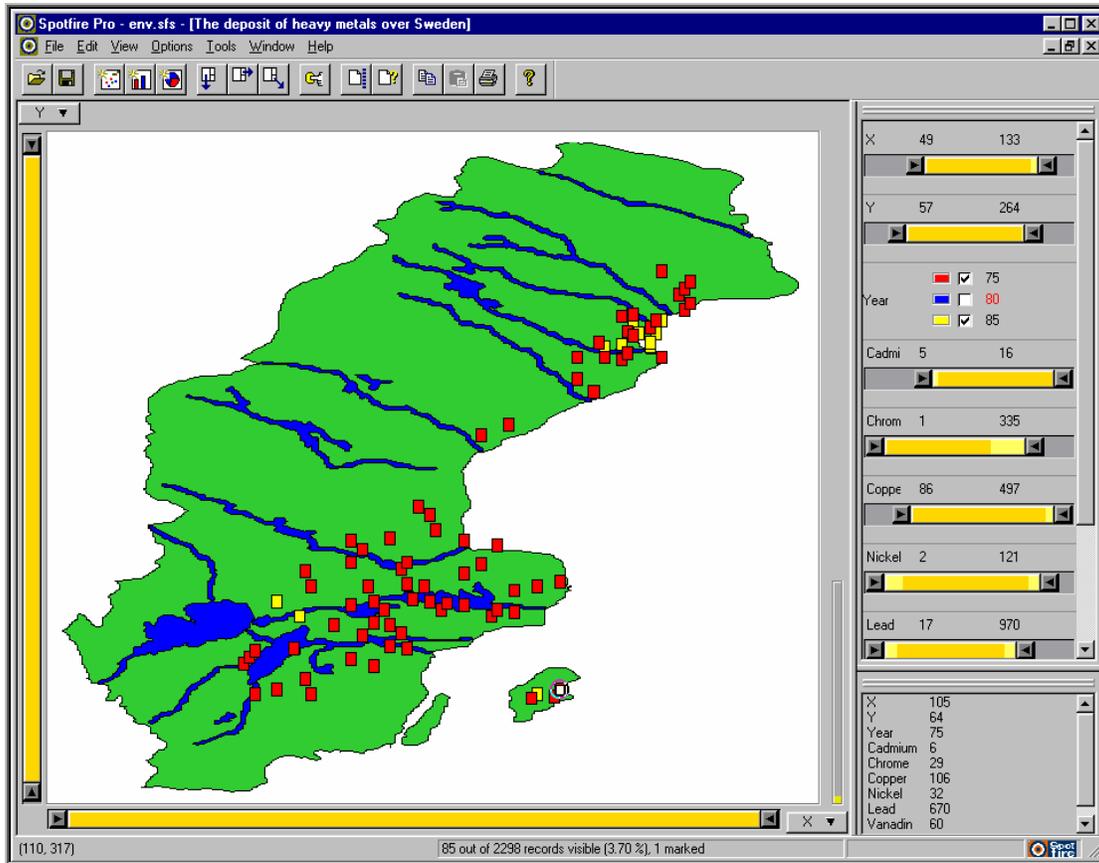


Figure 1. A sample dynamic query interface: Spotfire (www.spotfire.com) showing the distribution of heavy metals over Sweden. The widgets on the right, as well as the coordinate menus on the top left and bottom right, are used to define queries. The data is presented in a starfield display, here a map of Sweden. The total number of hits is displayed at the bottom of the interface. As users manipulate the sliders, results are continuously updated in the starfield display. Details of a selected item from the starfield are shown in the bottom right window.

While dynamic queries require all the attribute values of every record of the database to be downloaded from the network, query previews only need aggregate information about the data. The distribution information is represented as a multidimensional table that can be kept in the main memory of both the client

and the server computers. Each cell of this table represents a count of the database records corresponding to that cell of the table. The size of the table is independent of the size of the database. The counts are incremented (decremented) with each insertion (deletion) to the database. Only a few pre-selected attributes

of the data are used in the query preview interface. Thus, this table can be easily downloaded over the network.

Figures 2 and 3 show a query preview interface using the three most commonly used attributes of the Global Change Master Directory (topic, time, and area). The distribution of data over these attributes is shown with bar charts and the result set size is displayed in the result bar at the bottom.

### Query refinement

If needed, the query preview phase can be followed by a query refinement phase, which may be implemented as a dynamic query interface, to further refine the query. At the refinement phase, when a desired final result set size is obtained, the results can be retrieved from a (remote) database. Other types of user-interfaces for the refinement phase are also possible (i.e. form fillin, menus, etc).

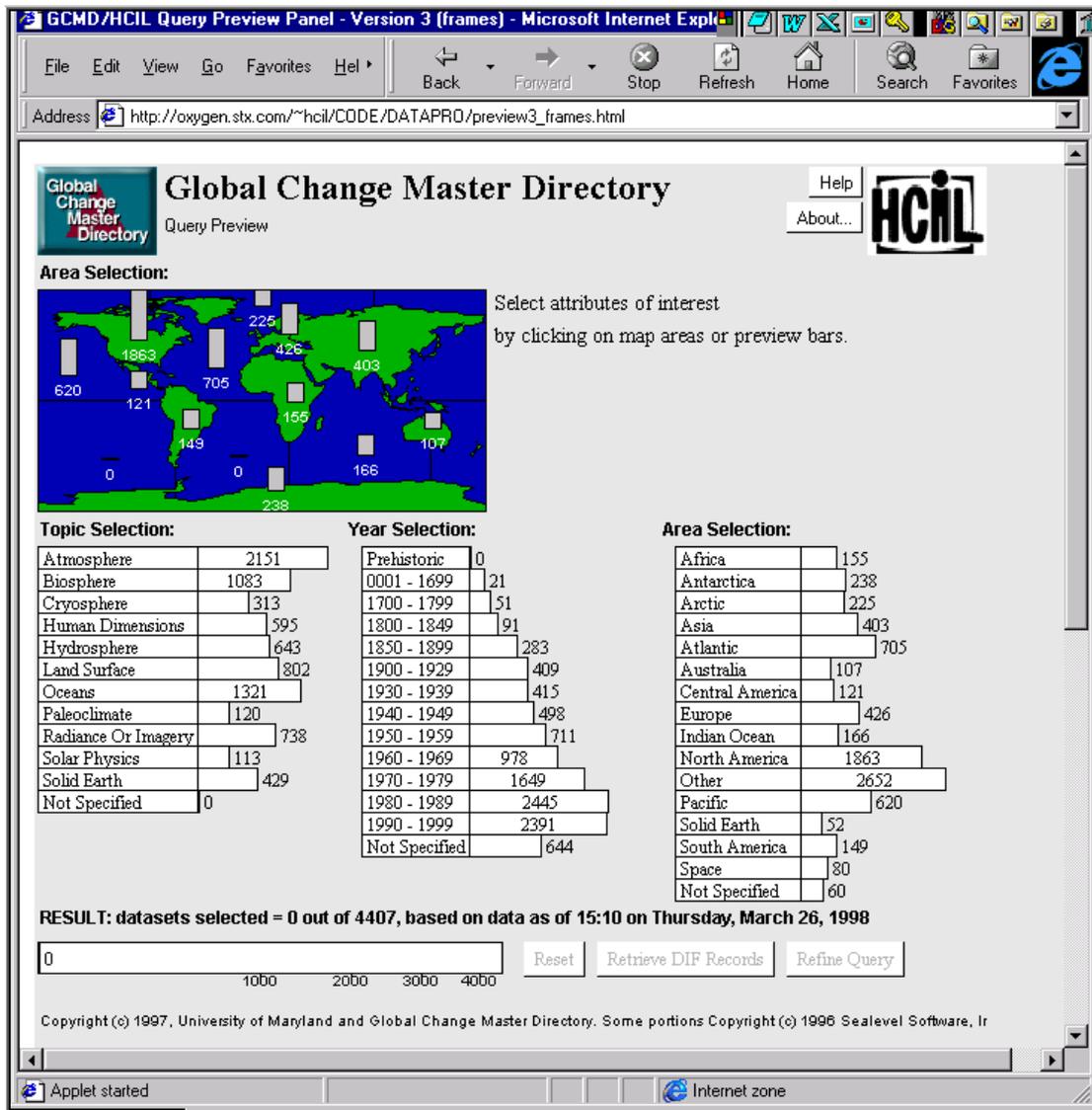


Figure 2. An example query preview developed at the Human-Computer Interaction Laboratory, for NASA Global Change Master Directory. Topic, Year, and Area are the discriminating attributes for the 4407 scientific data sets of the NASA archives. In this screen shot, the bars show the overview of the data distribution. Recent versions of this interface are available at the Global Change Master Directory: [gcmd.nasa.gov](http://gcmd.nasa.gov).

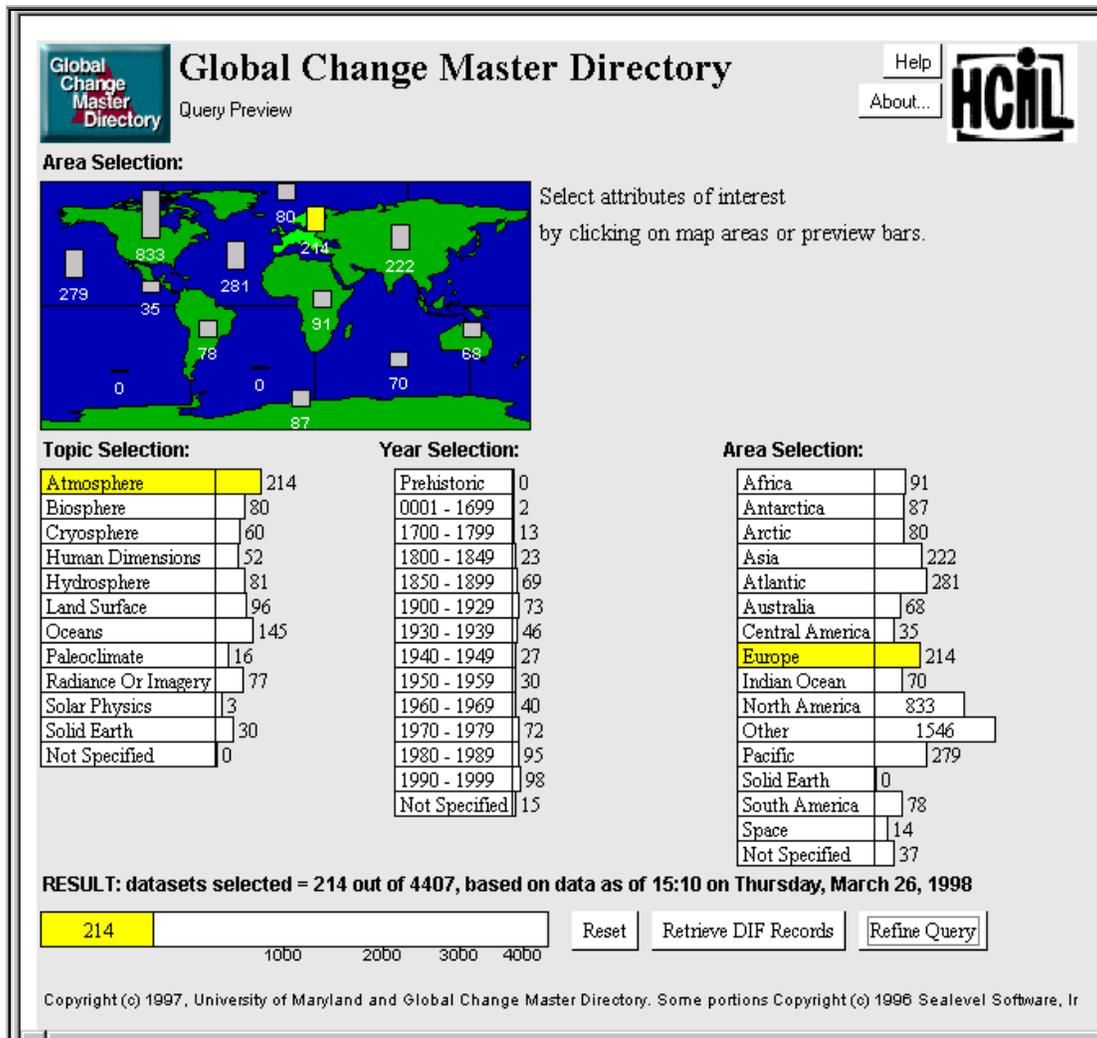


Figure 3. When users select attribute values (e.g. here atmosphere for topics and Europe for area), the bars are updated immediately to reflect the new distribution of the data that satisfies the query. When users are satisfied with their initial query, the results can be retrieved, or the query can be refined with additional attributes. In this case, atmosphere data for Europe produces a set with 214 datasets.

### 1.2. Motivation for empirical study

Since query previews add another phase to query formulation, there is a possibility that user performance would deteriorate and that users would be annoyed by a two-phase approach. Also, query previews focus attention on only a few selected attributes that may not be useful in some queries. During our demonstrations at NASA or to dozens of colleagues and visitors, it seemed obvious to enthusiastic observers that the preview was useful. They would often also rightly observe

that the query preview is not useful at all times (e.g. a string search or form fillin is probably best when you know the name of what you want). This study attempts to verify and quantify the benefits of query previews and measure the subjective user preferences. Our hope is that the query previews guide users in forming queries and will help them rapidly narrow down the search space to a manageable size.

Before the study, it became apparent that the variations in network delays could introduce a problem. Therefore, for better control, our

study was conducted in a non-networked environment. This corresponds to the worst case for query previews and any time advantage of query previews in our study corresponds to a larger advantage in a networked environment with delays. Section 2 describes the study, and Section 3 gives the results. Section 4 discusses the outcomes. Section 5 concludes with suggestions for researchers and practitioners.

## 2. User study methods

### 2.1. Introduction

We identified task types that would put query previews into their best and worst situation so that we can quantify the maximum benefits and drawbacks of this technique.

Clearly specified tasks: Clearly specified tasks have a straightforward and an accurate definition (known-item search), e.g. 'List all the Maryland employees from the employee database'. Query previews have no benefits for this task. In this case, users want to see the result list regardless of the type of this list. For this case and in general for clearly specified tasks the relevance of the query preview attributes to the query is not an influential factor since users are best served by going directly to the form fillin interface (Tasks of this worst case scenario are named as T1 tasks).

Unclearly specified tasks usually require a series of submissions. User's constraints and preferences cannot be stated immediately. Information gained from the query previews will influence their series of choices, so query previews could be very useful. But the relevance of the attributes used in the query preview will impact the usefulness of the user interface. Imagine that a user is looking for some software engineers from the Washington, DC area using an employee database. If the query preview shows the number of employees per state and some other attribute values of the database such as the age distribution, then the preview is only partially relevant to the task (middle case

scenario: T2). On the other hand, if the query preview shows the number of employees per state and their job types, then the query preview becomes fully relevant (best case scenario: T3). Because of the growth of public access to online databases, users often confront large amounts of data that is novel to them, in terms of content, size, distribution, data attributes, etc. For example, users of the US Census Bureau and the IBM patent databases are unlikely to know the years covered or the dominant themes of the data. The task types T2 and T3 target to represent such exploratory queries.

The three task types in the study varied in terms of their clarity of the specifications they have and in terms of their degree of relevance of the attributes they used to the query preview attributes. In this study, twelve subjects performed this varied set of tasks, once by using an interface that included a query preview and followed by a form fillin interface and once by only using a form fillin interface. The task completion times and the subjective preferences of the twelve subjects were measured.

### 2.2. Hypothesis

Our hypotheses were: (1) For clearly specified tasks (T1), adding the query preview step will lead to slower task performance (2) For unclearly specified tasks (T2 and T3), the addition of a query preview step will lead to faster performance. (3) Users will always prefer query preview interfaces.

The independent variable was the user interface type and the treatments were:

- Form fillin interface with a query preview.
- Form fillin interface without a query preview.

We examined the two interfaces using the three types of tasks that are defined as:

- T1: Clearly specified tasks in which the query preview attributes are not relevant to the task.
- T2: Unclearly specified tasks in which some of the query preview attributes are relevant to the task.
- T3: Unclearly specified tasks in which all of the query preview attributes are relevant to the task.

The dependent variables were the time to complete the tasks in each interface (not including setup times) and the subjective preferences of the users.

### 2.3. Subjects

Twelve computer science graduate students were used as subjects. All of them use computers almost every day and have at least five years of experience in using them. All, except one, stated that they regularly use Internet/database searching tools.

### 2.4. Materials

The materials include a form fillin interface for querying a film database (including 500 films), a query preview panel for the same database, a set of tasks to be performed by the subjects, a subject background survey, and a subjective user preference questionnaire.

#### Form fillin interface

The form fillin interface (Figure 4) is used to perform queries on a film database. There are ten attributes for a film in our sample database: category (horror, action, comedy, etc.), award winner (yes or no), rating (R, PG-13, PG, and G), year of production, length, popularity, lead actress, director, lead actor, and title. The output of a query is a list of films matching the specifications of the query. Vertical and horizontal scroll-bars can be used for scanning the list.

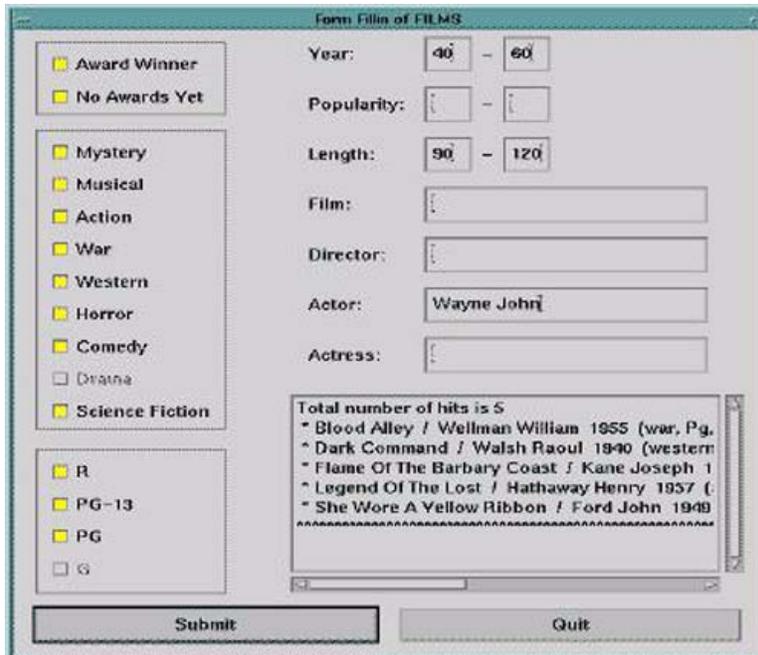


Figure 4. The form fillin interface used in the study. The rectangle on the right bottom corner is used for displaying the result list to a query. The list of fields allows users to enter values for the attributes of the database. The three attributes on the left side are the ones that are also available in the query preview.

### Query preview

In the query preview panel (Figure 5) users select values for three attributes of the database: the category (horror, action, comedy, etc.), whether the film won an award or not, and the rating (R, PG-13, PG, and G). Multiple selections are available for each of these attributes. The number of films for each attribute value is shown on a separate preview bar. Each preview bar consists of a frame and an internal rectangle (gauge). The length of the frame is proportional to the number of films in the database that match this specific value of the corresponding attribute. The length of the gauge is proportional to the portion of the films that match the query specified (the number of matches appears to the left of the bar). Users formulate queries by selecting the attribute values. As each value is selected, the preview bars of the other attributes adjust to reflect the

number of films available for those specific values (this is called tight coupling). For example, users might be interested only in films that won awards. By selecting 'Award Winners', the gauges of the preview bars of the selected categories and ratings change immediately to reflect only films with awards. The query preview bar at the bottom of the screen changes its length to illustrate the total number of films that match the current conditions. When the 'Refine' button is pressed, the query preview submits the specified partial query to the search engine and all the data about films that satisfy the query are downloaded for the query refinement phase. The query preview is closed and the form fillin interface is loaded to refine the query (displaying initially all the films selected in the query preview phase in the result box).

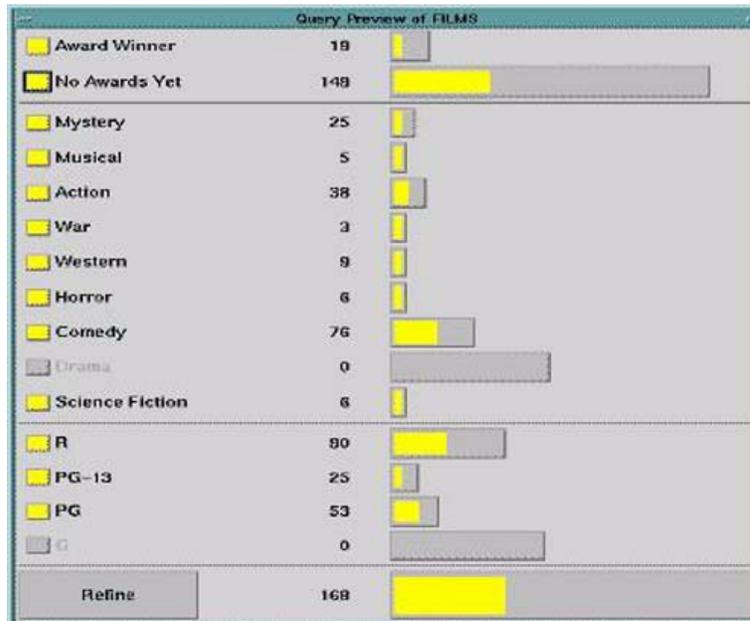


Figure 5. The query preview used in the study. The toggles on the left are used to choose attribute values to form the query. The counts and bars show the distribution of the result set for the query corresponding to the current settings of the toggles. The larger bar at the bottom shows the total number of hits, here 168.

## Tasks

The tasks given to the subjects were to find a film or a list of films in the database satisfying the constraints that we provided. Three types of tasks were used for this purpose:

- T1: a clearly specified task in which none of the query preview attributes is relevant for the task, e.g. 'Find the latest film by Alfred Hitchcock' (known-item search). For that type of task, users can typically find the answer by submitting a single form fillin query. The query preview has no advantage and its attributes are not relevant to the query.
- T2: desired films are vaguely specified. In this type of task, some of the query preview attributes are relevant, e.g. 'Find a PG-13 musical which was produced between years 1990 and 1995, if no such film is available, find a war film from the same years with the same rating, if not, try a musical or a war film from 1970-1990, and as the last possibility, try a comedy from 1970-1995'. This type of tasks is typical when users have a complex set of acceptable results, with some preferences. To perform such a search in the form fillin interface users must issue several queries, i.e. when the preferred choice is not available in the data. In the query preview, users can get some insight about what is available in the database and what is not and hence can make more informed queries. However, since not all the attributes of the query specification appear in the query preview interface, the form fillin interface is required to refine the query.
- T3: formed in a similar way to T2. A series of preferences for films are specified. In this case however, the query preview attributes are fully relevant to the task specifications. Example: 'Find at least 30 films of the same category which are R rated and have no awards' (for example, in order to organize a

film festival or make a collection). In the form fillin interface this task requires several queries to examine the number and rating of films in each category. The query preview on the other hand, gives an immediate picture of the relevant categories. The form fillin interface is required only to get an explicit list of the films.

For each of the above task types, six examples were prepared (18 tasks in total).

### *Subject background survey and preference questionnaire*

The survey included 8 questions that ascertained the experience level of the subjects with computers and with search engines. We also prepared a preference questionnaire. The subjective user preference questionnaire included 6 questions that aimed to find out which of the two interfaces (a form fillin with or without a query preview) the subjects preferred and what their attitudes were toward adding query previews to the querying interface.

### *2.5. The user study design*

The study used a within subject counter-balanced design with 12 subjects. Each subject was tested on both of the interfaces, but the order of the interfaces was reversed for half of the users. A parallel set of tasks (similar but not the same set of tasks) was used on the second interface to reduce the chance of performance improvement. Each set of tasks included the three types of tasks (T1, T2, T3), with three tasks for each of these types. The order of the task types within a task set was also reversed (each of the six permutations was used twice). The order of the tasks within each task type was fixed.

## 2.6. Procedure and administration

The subjects signed a consent form, filled out a background survey, received a brief demo of the form fillin interface and the query preview, and a 10 minute training session in which they used the two interfaces (with similar but not same tasks to the actual tasks that were used). During the study each subject performed 18 tasks (9 in each of the interfaces). At the end of the study the subjects filled in the preference questionnaire. The study took 50-60 minutes including the training and the questionnaires.

Two administrators were present. One administered the study, performed the demo, presented the tasks, and measured the task execution times. The other administrator recorded notes about the way subjects coped with the tasks and about problems that may occur during the study, and verified the procedures that were followed. The time that the subjects spent in using each of the interfaces was recorded: successful completion time of a task. These completion times did not include program startup times. The clock was started after the subjects read the question and the user-interfaces became available on the computer

screen. When the subjects found the answer for the query and showed the result to the administrator, the clock was stopped.

## 3. Results

### 3.1. Time for completing the tasks

Figure 6 summarizes the times for completing each of the task types for our subjects (clearly specified: T1, unclearly specified and partially relevant: T2, unclearly specified and fully relevant: T3) for each of the user interfaces (with and without a preview). For T1 tasks the user interface with the query preview yielded slower performance than the user interface without a query preview ( $t(35) = 2.44$ ,  $p < 0.05$ ). For T2 and T3 tasks the interface with the query preview yielded faster performance than the interface without a query preview ( $t(35) = 8.77$ ,  $p < 0.05$ , and  $t(35) = 14.70$ ,  $p < 0.05$ , respectively). The statistical analysis used one-tailed paired two-sample t-test for means. Each task is considered separately leading to degrees of freedom of 35.

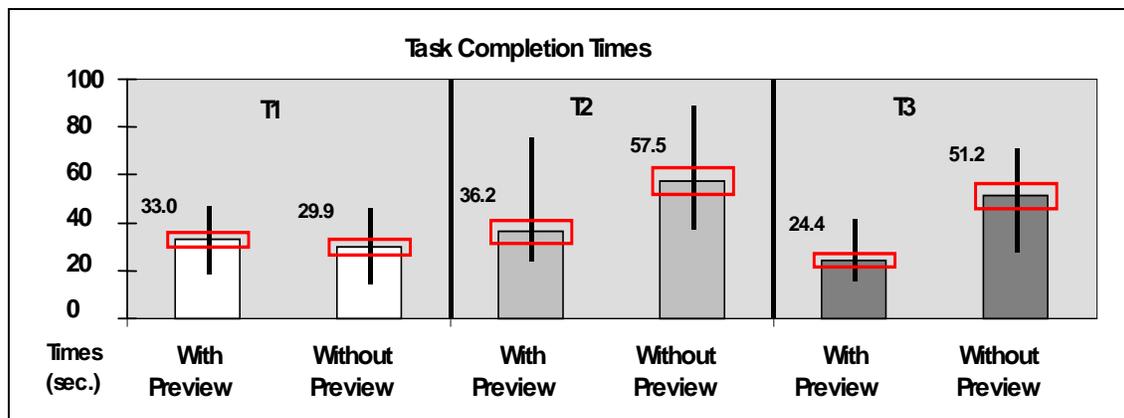


Figure 6. Average task completion times for T1, T2, and T3 (the rectangles show the standard deviations and the vertical lines indicate the ranges).

### 3.2. Subjective satisfaction

The subjects answered six questions about their preferences on a 1 to 9 scale (with higher numbers indicating stronger preferences). The first question examined the general preference of subjects for using a form fillin interface with or without a query preview (Figures 7a and 7b). The results showed a statistically significance difference ( $t(11) = 2.82, p < 0.05$ ) for the interface with a query preview over the interface

without a query preview. The rest of the questions asked what the subjects thought about the user interfaces. The results (average scores, standard deviations, minimums, and maximums) appear in detail in Figure 8. The scores for all of the questions were statistically significantly above the mid-point scale value of 5.0 ( $t(11) = 3.86, 6.20, 7.71, 2.24, \text{ and } 2.58$  respectively,  $p < 0.05$ ).

Which of the user interfaces you liked the most?

a) The one WITH the query preview (please indicate on the scale):

Dislike									Like
1	2	3	4	5	6	7	8	9	

b) The one WITHOUT the query preview (please indicate on the scale):

Dislike									Like
1	2	3	4	5	6	7	8	9	

Figure 7a. The subjective user preference question that was used in the study.

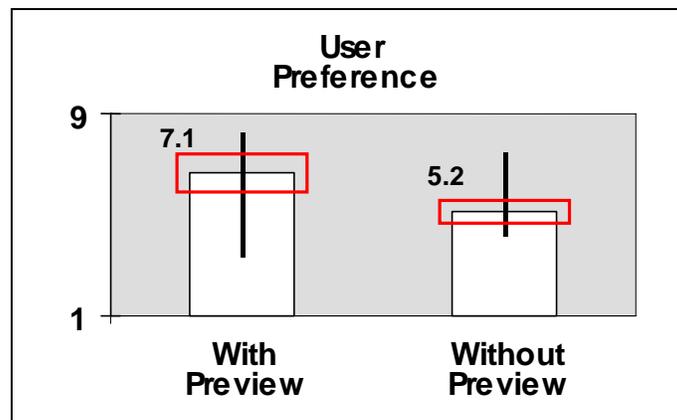


Figure 7b. The subjective user preference results for the 12 users.

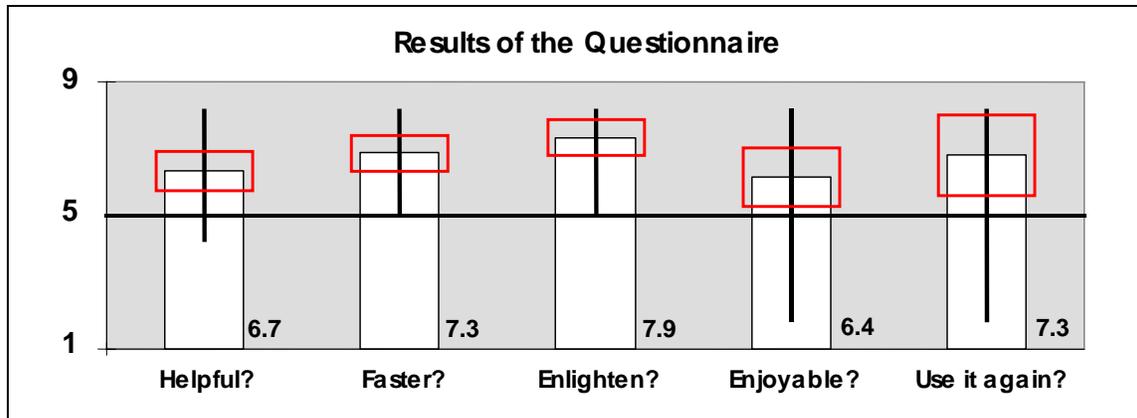


Figure 8. The rest of the subjective user preference questionnaire results for the interface with the query preview. Higher numbers indicate higher satisfaction for using the query preview.

#### 4. Discussion

Our findings support the hypothesis that for unclearly specified tasks, the interface with a query preview yields better performance times than the interface without a query preview. For both types of the unclearly specified tasks the improvement in performance was significant (at the level of 0.05): 1.6 times faster for T2 tasks and 2.1 times faster for T3 tasks. For the clearly specified tasks (T1), as expected, the form fillin only interface performed slightly better.

##### 4.1. Clearly specified tasks (T1)

As expected, users of the form fillin interface for clearly specified tasks performed more rapidly since they were able to find the answer by submitting a single form fillin query. The query preview had no advantage and its attributes were not relevant to the query. Users performed simple known-item searches. However, users of the interface with the query preview performed only slightly worse (10% slower). The users spent 2-3 seconds in the query preview, identified that its attributes are not relevant or needed for the task and continued to the refinement phase.

##### 4.2. Unclearly specified tasks, with partial relevance of the query preview attributes (T2)

Although not all the attributes of the task specification could be specified using the query preview, the insight gained from the query preview enabled users to eliminate some potential zero hit queries in advance, concentrating, in the refinement phase, on a much smaller set of possible queries. The query preview enabled the users to reduce the search space significantly so that they could find the answer more quickly.

##### 4.3. Unclearly specified tasks, with full relevance of the query preview attributes (T3)

For unclearly specified tasks with full relevance of the query preview attributes, the full power of the query preview was used. The query preview enabled the users to see immediately which of the possible queries should be submitted. The users loaded the refinement phase only for submitting the query and viewing the results. The users performed the refinement phase with a high confidence that they would get the expected results. On the other hand, in the user interface without a query preview, the users had no clue about which of the possible queries will give the expected results. They had to try several possible queries

until they got a satisfactory answer. Although the response time for each such query was immediate, the time for finding and filling in the right specifications of each query caused the significant differences in performance (even more than T2's).

#### 4.4. *Performance improvement*

The results show that for different types of tasks the query preview achieves different rates of performance improvement in comparison with the traditional form fillin interface (from 0.1 times slower in T1 to 2.1 times faster in T3). The performance improvement depends on several parameters. One parameter is the clarity of the task specifications. In clearly specified tasks there is almost no potential for improvement. Another important parameter is the relevance of the query preview attributes to the task. Two additional parameters are the significance of the query preview attributes in pruning the search space and the resolution of the attribute values. For example, if rating R is used and almost all the films in the database are of rating R, this attribute, although relevant, has insignificant contribution to the performance improvement. When numeric attributes such as year of production or length of the film are presented in a query preview, the possible values for these attributes are presented using some pre-defined resolution (for example, a 10-year resolution). Tasks that require higher resolution for an attribute than the one provided in the query preview will gain less benefit from the query preview.

In our study, the query preview yielded more performance improvement for T3 tasks (full relevance of the query preview attributes) than for T2 tasks (partial relevance of the query preview attributes). This result might support the assumption that better relevance of the query preview attributes to the task yields more performance improvement.

#### 4.5. *Learning to use a query preview*

We found that it was easy for users, with experience in querying a database using the form fillin interface, to learn the query preview interface and take advantage of the information it supplies. However, some of the users, during training and, in few cases, during the study, continued with the refinement phase immediately, skipping the examination of some of the relevant attributes. That happened when not all the task attributes could be found in the query preview. For example, when performing a task with conditions on rating (in the query preview), year (not in the query preview) and category (in the query preview), the fact that the year could not be specified in the query preview caused some of the subjects continue to the refinement stage without examining the information for the category attribute.

#### 4.6. *Subjective satisfaction*

The users preferred the interface with the query preview, over the interface without it. They stated that the query preview was helpful, enabling them to search faster, and learn more about the database (scores for these questions were statistically significantly above the mid-point value). We believe that this subjective satisfaction comes not only from the improvement in performance time which is experienced by the subjects but also from gaining better control in performing the tasks.

The suggested improvements related to the user interfaces are: supplying a way to clear a group of related check boxes in one step, or easily resetting or setting all of them, a more immediate refresh operation on the preview bars for visual accuracy when changing the attribute values of the query preview panel. The significant preference that subjects showed for including query previews in search engines they currently use (in addition to the objective performance improvement for two of the task types), encourages further efforts in

understanding, refining, and developing query preview interfaces.

## 5. Conclusions

### 5.1. *Impact for practitioners*

This user study shows that query previews are powerful tools for exploring databases. Query previews give insight about the database that is being searched and guide users throughout the query formulation process.

Tasks that have unclear definitions generally lead to longer task completion times in form fillin interfaces. Query previews are very useful in these situations. The benefits obtained depend highly on the relevance of the query attributes to the attributes used in the preview. The costs introduced by the preview are negligible with respect to the benefits (e.g. short delays in query preview load time, implementation costs, and extra training for the preview interface).

We observed that tasks that have a clear definition (regardless of the relevance of the tasks to the query preview) were easily executable on a regular form fillin interface. Query previews were not needed in these situations and, as anticipated, they introduced small delays. On the other hand, delays that were introduced by the query preview panel did not seem to annoy the users (due to the fact that they were relatively short delays).

With typical delays in networked environments, we expect greater benefits from a query preview than the ones we observed in this study. Besides, as the size of online databases get larger, we think that the benefits of query previews will be more appreciated by the users.

Most of the users preferred the query preview. This is probably due to the fact that users gain greater control and insight about the database while using a preview. Viewing the data distribution over the whole space of records was very helpful for the users. The immediate feedback that was given to users was also found

to be very useful. However, relevance of the preview attributes to the most commonly used attributes should be high to maximize these benefits.

### 5.2. *Suggestions for researchers*

We suggest that future studies explore:

- the effect of network delays,
- the benefits of query previews in conjunction with menus or visual interfaces,
- other task types (with more varied relevance levels),
- a variety of users (novices, experts, etc),
- the definition of a concrete measure of clearness and relevance of the preview attributes to the query.

### 5.3. *Conclusions*

This study is the first user study done on query previews. Our recent work with the Global Change Master Directory suggests that query previews are feasible in operational networked systems. Query previews are not proposed as a useful technique for all query interfaces and types of tasks but this first study confirms that the benefits of query previews exist for several tasks, even in non-networked environments. The study shows that task types play a critical role in performance improvements. A categorization of tasks for exploring with query previews was introduced (clear vs. unclear, relevant vs. irrelevant) but more precise measures for clearness and relevance might be useful in future studies. Future research should investigate other aspects of query previews such as: a quantification of the network access reduction, and a better understanding of the benefits of the query previews on the incidental user knowledge of the database contents.

## Acknowledgements

This work was partially supported by NASA (ESDIS) grant NAG 52895.

## References

- Ahlberg, C. and Shneiderman, B., 1994, Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays, *Proceedings of the ACM CHI '94 Conference*, ACM, New York, 313-317.
- Doan, K., Plaisant, C., and Shneiderman, B., 1996, Query Previews in Networked Information Systems, *Proceedings of the Forum on Advances in Digital Libraries*, IEEE Society Press, 120-129.
- Eick, S., 1994, Data Visualization Sliders, *Proceedings of ACM UIST '94 Conference*, ACM, New York, 119-120.
- Goldstein, J. and Roth, S.F., 1994, Using Aggregation and Dynamic Queries for Exploring Large Data Sets, *Proceedings of the ACM CHI '94 Conference*, ACM, New York, 23-29.
- Greene, S., Tanin, E., Plaisant, C., Shneiderman, B., Olsen, L., Major, G., Johns, S., 1999, The End of Zero-Hit Queries: Query Previews for NASA's Global Change Master Directory, *International Journal of Digital Libraries*, 2, 2, 79-90.
- Hearst, M., 1995, Tilebars: Visualization of Term Distribution Information in Full Text Information Access, *Proceedings of the ACM CHI '95 Conference*, ACM, New York, 59-66.
- Heppe, D., Edmondson, W.S., and Spence, R., 1985, Helping both the Novice and Advanced User in Menu-driven Information Retrieval Systems, *Proceedings of HCI '85 Conference*, British Computer Society Press, 92-101.
- Nation, D.A., Plaisant, C., Marchionini, G. and Komlodi, A., 1997, Visualizing Websites Using a Hierarchical Table of Contents Browser: WebTOC, *Proceedings of the 3rd Conference on the Human Factors and the Web*, <http://www.uswest.com/webconference/proceedings/nation.html>.
- North, C., Shneiderman, B., and Plaisant, C., 1996, User Controlled Overviews of an Image Library: A Case Study of the Visible Human, *Proceedings of the 1st ACM International Conference on Digital Libraries*, ACM, New York, 74-82.
- Plaisant, C., Bruns, T., Doan, K., and Shneiderman, B., 1999, Interface and Data Architecture for Query Previews in Networked Information Systems, *ACM Transactions on Information Systems*, ACM, New York, 17, 3, 320-341.
- Shneiderman, B., 1994, Dynamic Queries for Visual Information Seeking, *IEEE Software*, 11, 6, 70-77.
- Shneiderman, B., Byrd, D., and Croft, W.B., 1997, Clarifying Search: A User-Interface Framework for Text Searches, *D-Lib Magazine*, January, <http://www.dlib.org/dlib/january97/retrieval>.
- Veerasamy, A., and Navathe, S., 1995, Querying, Navigating and Visualizing a Digital Library Catalog. *Proceedings of the Second International Conference on the Theory and Practice of Digital Libraries*, <http://www.cSDL.tamu.edu/DL95>.
- Williams, M.D., 1984, What Makes RABBIT Run, *International Journal of Man-Machine Studies*, 21, 4, 333-352.
- Williamson, C. and Shneiderman, B., 1992, The Dynamic Home Finder: Evaluating Dynamic Queries in a Real-Estate Information Exploration System, *Proceedings of ACM SIGIR '92 Conference*, ACM, New York, 338-346.