Query Previews for Networked Information Systems: A Case Study with NASA Environmental Data

Khoa Doan[*], Catherine Plaisant, Ben Shneiderman[#], and Tom Bruns
Human-Computer Interaction Laboratory
Institute for Advanced Computer Studies
Department of Computer Science and Institute for Systems Research
University of Maryland, College Park, MD20742
e-mail:plaisant,ben@cs.umd.edu
hhtp://www.cs.umd.edu/projects/hcil

ABSTRACT

Formulating queries on networked information systems is laden with problems: data diversity, data complexity, network growth, varied user base, and slow network access. This paper proposes a new approach to a network query user interface which consists of two phases: query preview and query refinement. This new approach is based on dynamic queries and tight coupling, guiding users to rapidly and dynamically eliminate undesired items, reduce the data volume to a manageable size, and refine queries locally before submission over a network. A two-phase dynamic query system for NASA's Earth Observing Systems--Data Information Systems (EOSDIS) is presented. The prototype was well received by the team of scientists who evaluated the interface.

Keywords: User interface, direct manipulation, dynamic query, metadata, query preview, query refinement, EOSDIS.

INTRODUCTION

The exploration of networked information resources becomes increasingly difficult as the volume of data grows and as the complexity increases. Congested networks and a varied user population contribute to the problems of information retrieval.

In this paper, we present a case study showing auser interface to support efficient query formulation for networked information systems using dynamic queries and query previews [DPS97]. The case study is based on our work with NASA's environmental data.

Dynamic query user interfaces have been developed at the Human Computer Interaction Lab for several years [Shn94]. Dynamic query user interfaces apply the principles of direct manipulation to query formulation:

- visual representation of the query
- visual representation of the results
- rapid, incremental, and reversible control
- selection by pointing, not typing
- immediate and continuous feedback

Dynamic queries involve the interactive control by a user of visual query parameters that generate a rapid (under 100msec), animated, and visual display of database search results. As users adjust sliders or buttons, results are updated in real time on the display.

The enthusiasm users have for query previews emanates from the sense of control they gain over the database. Empirical results have shown that dynamic queries are effective for novice and expert users to find trends and spot exceptions [Will93].

Early implementations of dynamic queries used relatively small datasets of a few thousand datapoints as they required the data to be stored in memory to guarantee rapid update of the display. We are working on algorithms and data structures that support larger datasets (up to 100,000 datapoints) [Tan96], but slow network performance and limited local memory remains an obstacle when trying to use dynamic queries for very large distributed datasets. Query previews offer a solution to this problem.

QUERY PREVIEWS

Query previews combine browsing and querying. Summary data about the database (such as the number of datasets in pre-defined categories) are used to guide users to reduce the scope of their queries and to focus only on the datasets of interest. The summary data is generally orders of magnitude smaller than the database itself, and can be downloaded from the server quickly to drive a dynamic query interface locally on the user's client machine.

---

[*]Current Address: Khoa Doan, Hughes STX Corp, 7701 Greenbelt Rd. Suite 400, Greenbelt MD 20770 , e-mail: kdoan@hops.stx.com

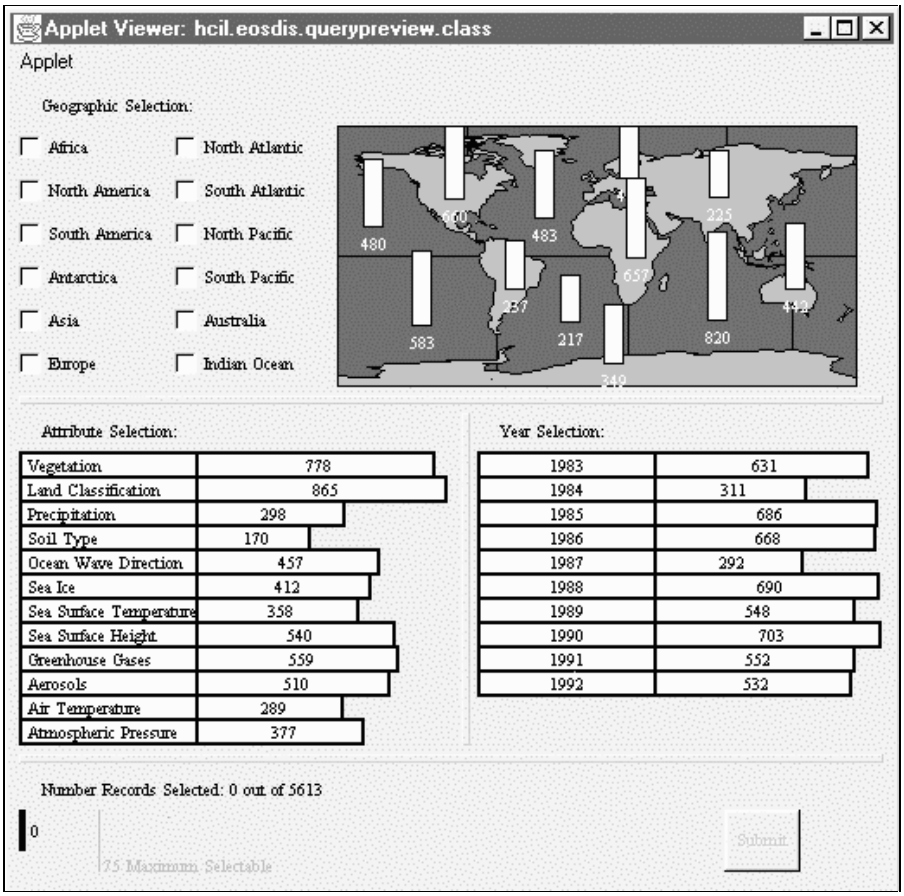

Figure 1: The query preview screen displays summary data on preview bars. Users learn about the holdings of the collection and can make selections over a few parameters (here geographic, environmental parameter and year).

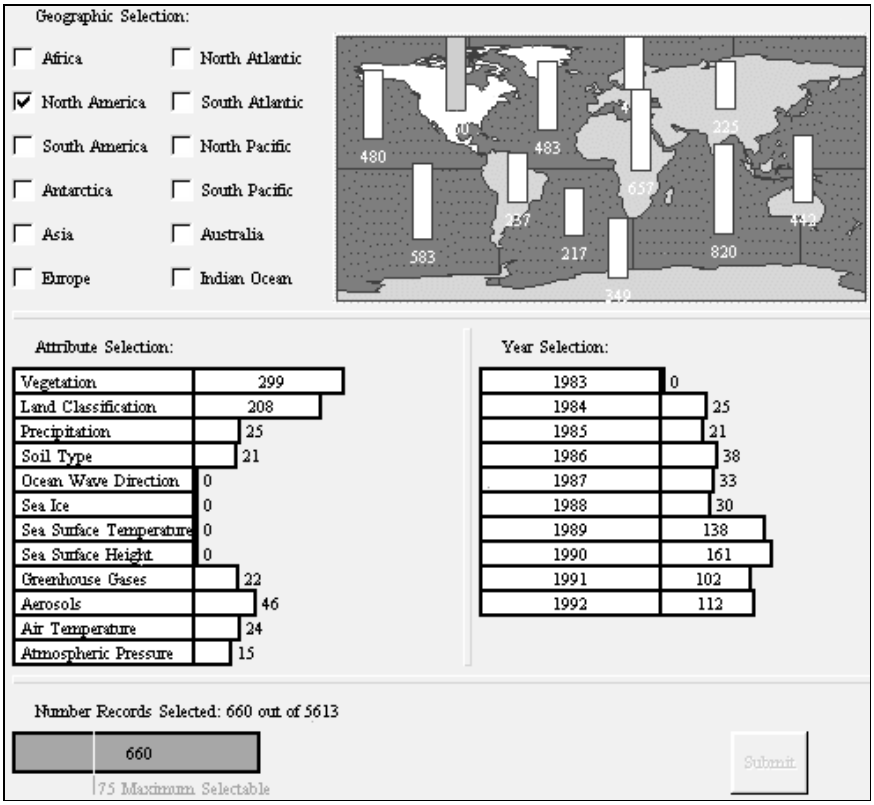

Figure 2: Following the principles of dynamic queries the preview bars are updated immediately (in less than 100msec.) when users select a attribute value (here North America). The result bar at the bottom shows the total number of selected datasets.

In the query preview phase, users form a rough query by selecting values over a small number of attributes, each of which has a small number of aggregated attribute values. The scope of the query is large, but the resolution is limited.

In the query refinement phase, users construct precise queries over all database attributes and values, which are applied only to those items selected in the query preview phase. The scope of the query is smaller, but the resolution is finer.

CASE STUDY AND PROTOTYPE

Our case study uses the NASA's Earth Observing System Data and Information System (EOSDIS) to illustrate our approach. Soon users (scientists, teachers, students etc.) will be able to retrieve earth science data from hundreds of thousands of datasets containing pictures, measurements, or processed data, from centers around the country. Data about the datasets (called metadata) is available and is used to search for useful datasets. Standard EOSDIS metadata includes spatial coverage, time coverage, type of data, sensor type, campaign name, level of processing etc.)

Keyword-oriented or form-based interfaces are widely used today for formulating queries on networked information systems and are available for EOSDIS. They often generate zero-hit queries, or query results that contain large number of datasets through which users still have to browse. Users can limit how much data a query should return (e.g. 20 "hits") to shorten of the search but it is then impossible to estimate how much data was not returned, and how representative of the entire search space the returned data was. Users also often fail to find data if appropriate keywords cannot be guessed.

A prototype of dynamic query preview was implemented in Tcl/Tk (video available from HCIL) and more recently a partial Java implementation was prepared to demonstrate the feasibility on the World Wide Web (WWW). The data shown in the current prototype are hypothetical, and we are now working with NASA to include real summary data.

EOSDIS QUERY PREVIEWER

In the query preview screen (Figure 1) users select rough ranges for a three attributes: geographical location (a world map with 12 regions is shown at the top of the screen), parameters (a list of parameters such as vegetation, land classification or precipitation), and temporal coverage (in the lower right of the screen). The spatial coverage of datasets are generalized into continents and oceans. The temporal coverage is defined by discrete years.

The number of datasets for each parameter, region, and year is shown on *preview bars*. The length of the preview bars is proportional to the number of the datasets containing data corresponding to the attribute value. At a glance we can see that the datasets seem to cover all areas of the globe but there is more data on North America than South America, and that parameters and years are covered relatively uniformly in this hypothetical EOSDIS dataset collection. The result preview bar, at the bottom of the interface, displays the total number of datasets. Note that only rough queries are possible since the spatial coverage of datasets are generalized into continents and oceans and the temporal coverage is defined by discrete years.

A query is formulated by selecting attribute values. As each value is selected, the preview bars in the other attribute groups adjust to reflect the number of datasets available. For example, a user might be interested only in datasets that contain data for North America, which are selected by clicking on the North America checkbox (left of the map) or by clicking on the image of North America on the map. The interface changes immediately (in few milliseconds) in response to this selection. The preview bars change to reflect the distribution of datasets for North America only. The query preview bar at the bottom of the interface changes size to illustrate the number of datasets selected by picking North America.

The user continues to define a preview query by selecting from other parameter groups (e.g. "Vegetation" and "Land Classification".) The preview bars in the spatial and year parameter groups adjust to reflect the new query (Figure 2), showing the number of datasets having vegetation or land classification data in North America.

The OR operation is used within attribute, the AND operation between attributes. Those AND/OR operations are made visible by the behavior of the bars which grow or shrink accordingly. Continuing, the user further reduces the number of selected datasets by choosing specific years (e.g. 1986, 1987, and 1988, three years which have data as shown on the preview bars)(Figure 3.)

When the "Submit" button is pressed the query previewer submits the specified rough query to the EOSDIS search engine and all the metadata of the datasets that satisfy the query are downloaded for the query refinement phase. In the example the query previewer had narrowed the search to 66 datasets.

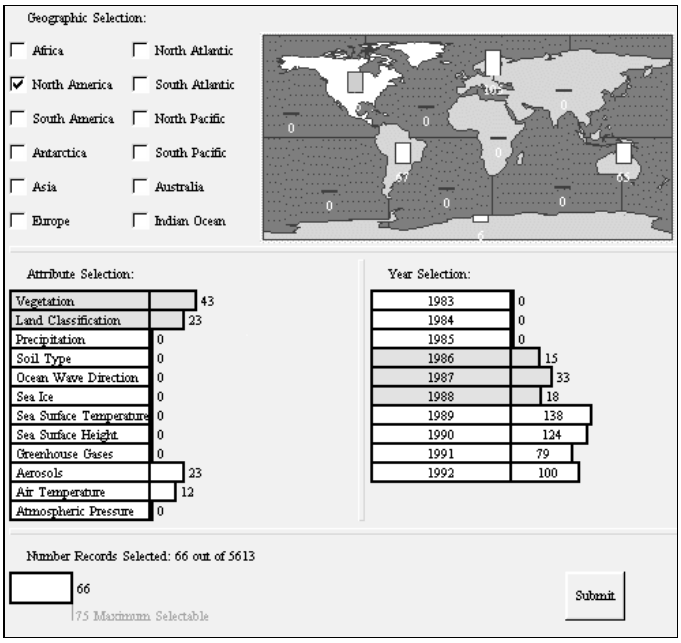

Figure 3: Three years (86 to 89), vegetation and land classification have been selected. All preview bars are updated. The query can now be submitted.

EOSDIS QUERY REFINEMENT

The query refinement interface supports dynamic queries over the metadata , i.e. over all the attributes of the datasets including: the detailed spatial extent and temporal interval, parameters measured in the dataset, the sensor used to generate the dataset, the platform on which the sensor resides, the project with which the platform is associated, the data archive center where the data is stored, and data processing level which indicates raw sensor data to highly processed data (levels 0 to 4).

A temporal overview of the datasets is given in the top left of the screen (Figure 4). Each dataset is now individually represented by a selectable line. At the bottom of the screen a table lists all the datasets and gives exact values for the attributes.

In the refinement phase of the query users can select precise values for the attributes. The map, already zoomed to the area selected in the query preview, should be zoom-able to allow precise selection. The time line of the overview, already narrowed to the years selected in the query preview can be re-scaled to specify narrower periods of interest.

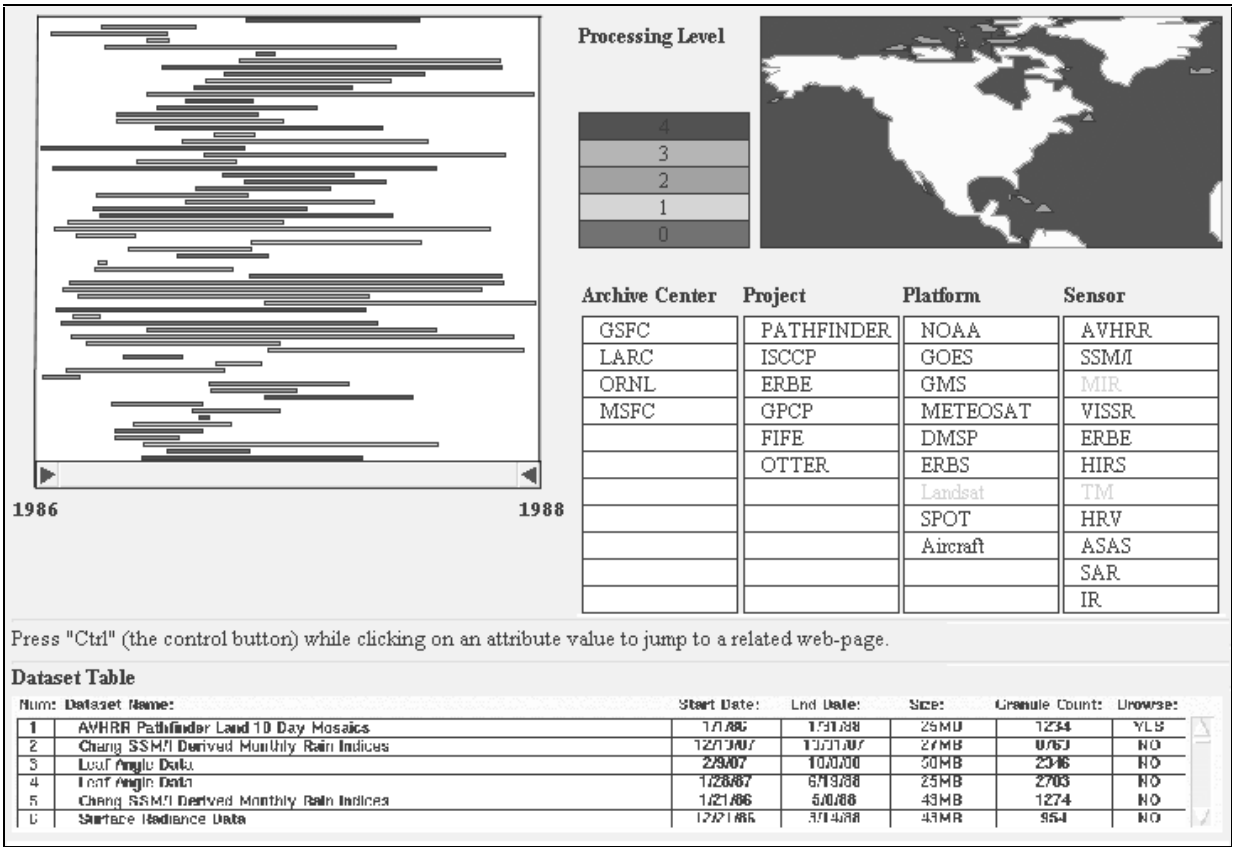

Figure 4: In the Query Refinement users can browse all the information about individual datasets. The result set can be narrowed again by making more precise selections on more attributes.

In this second dynamic query interface the result of the query is immediately visualized on the overview. As attribute values are selected the number of lines on the overview change to reflect the query in a few milliseconds since there is no access to the network.

All controls are tightly coupled to describe selected datasets (by showing attributes when a dataset is highlighted) and indicate valid values. In the example of Figure 5 the number of datasets was reduced by selecting the processing levels 2 and 3, two archive centers, and three projects. More details about a dataset such as descriptive information and sample data can be retrieved on demand from the network before the decision to download a full dataset is made.

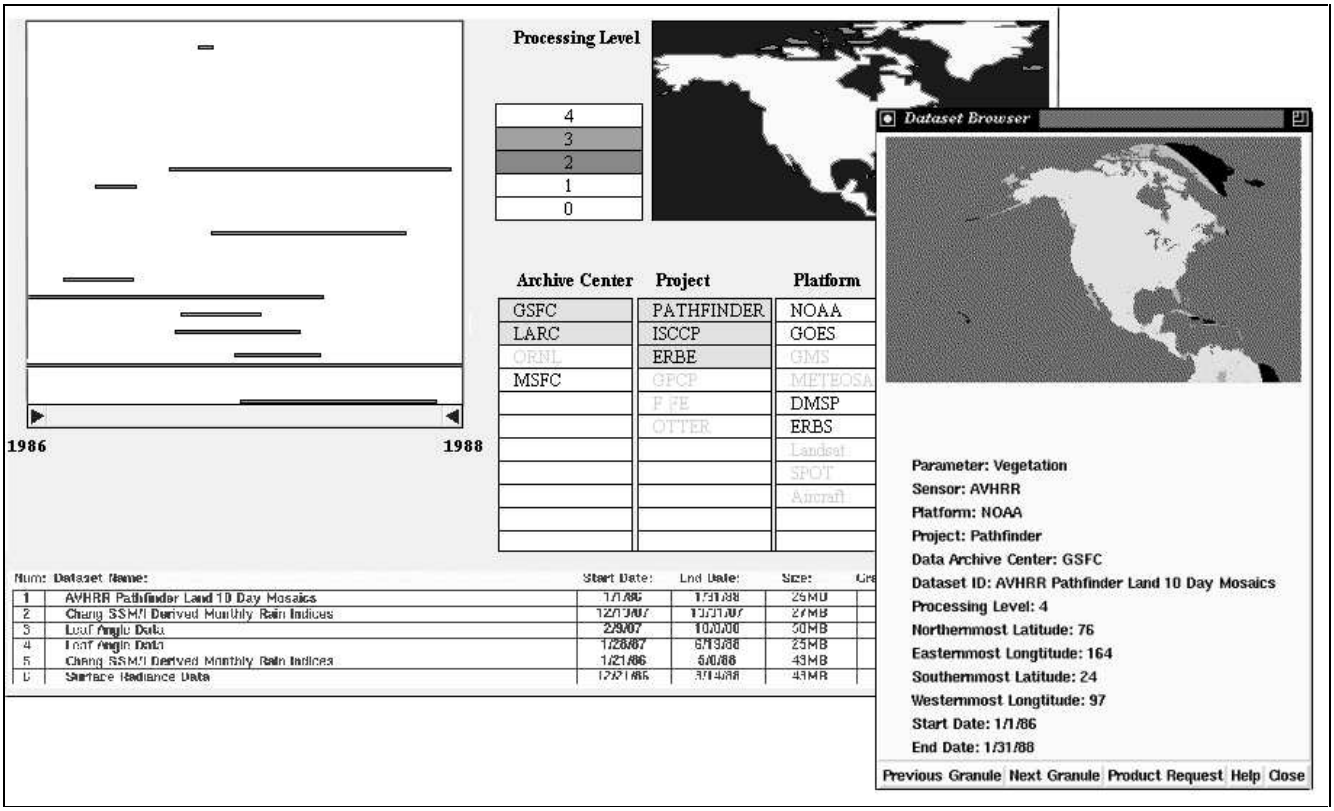

Figure 5: Here the query has been refined by selected 2 archive centers, 3 projects and 2 processing levels. More filtering could be done by zooming on the timeline or on the map. The timeline overview and the dataset table reflect the remaining datasets. Details and samples images can be downloaded from the network (window on the right) before the long process of ordering the large datasets.

VOLUME PREVIEW TABLE

The size and dimensionality of the volume preview table is a function of the number of preview attributes and the number of discrete preview values for each attribute. Consider a restaurant search application with three preview attributes: cuisine type, rating, and accepted credit cards. Imagine five types of cuisine, four ratings, and two acceptable credit cards. In the simple case where each restaurant's attribute only takes a single value the volume preview table would be a five-by-four-by-two table, with a total of 40 combinations.

N preview attributes, yield an N-dimensional volume preview table. The total size of the table is many orders of magnitude smaller than the size of the database, or the size of the dataset's metadata. Furthermore, the volume preview table does not change size as the database grows. The size of the volume preview table allows it to be loaded into local high-speed storage to support dynamic queries in the query preview phase.

Nevertheless, the number of attributes and the number of the possible values needs to be carefully chosen if the objects being searched (e.g. restaurants or datasets) can

take any combinations of values for their attributes. In the case of EOSDIS a given dataset can contain measurements of several parameters, covering several areas over several years. In the worst case (i.e. if all combinations are possible) the size of the preview table could become $2^{12}x2^{12}x2^{10}$ (for 12 areas, 12 parameters and 10 time periods) which would lead to megabytes of data, much too large to load over the network and use in the previewer.

A first solution is to ignore in some way the possible combinations and count twice the datasets that have 2 parameters, once in each cell for each parameter it contains. This will result in correct individual preview bars (e.g. the preview bar for 1990 really gives the total number of datasets that have any data for that year) but over inflate total result preview bar since some datasets are counted multiple time. This might be acceptable if combinations are a small proportion of the data, which is likely to be common because of the high granularity of the selections in the query previewer.

Another more accurate solution to the problem is to analyze the number of combinations, either by looking at the type of attribute (e.g. year combinations are typically year ranges, reducing the number of combinations to 55 instead of 1024 for 10 values), or because of the distribution of the data itself (e.g. EOSDIS parameters are grouped into only a limited number of compatible combinations).

The first solution has the advantage of keeping the size of the volume preview very small (e.g. 12x12x10 integers for our EOSDIS prototype, i.e. much smaller than the world map graphic), the second gives a more accurate preview but requires more time and space.

In our current prototype we chose to simply duplicate datasets because we did not have access to large amounts of real EOSDIS metadata. We are now working with the operation data center to select attributes and values ranges that will lead to reasonably sized preview tables.

Since the data of the networked information system changes regularly volume preview tables have to be updated. Our approach depends on the data providers being willing and able to produce and publish Volume Preview tables on a regular basis (weekly, daily or hourly depending on the application), or on third party businesses running series of queries to build the tables. Since the previewer is only meant to enter rough queries it is acceptable to use slightly out of date volume tables. The query previewer interface needs to make clear that the volume preview is an approximation on the real volume and give the “age” of the statistical information used. When the rough query is submitted, the (up-to-date) databases are queried and will return up-to-date data for the query refinement. At this point the number of dataset returned might be slightly different that predicted by the query preview. This might be a problem when the query preview predicts zero hit while a new dataset that would answer the query has just been added to EOSDIS. This risk has to be evaluated and adequate scheduling of the updates enforced.

RELATED WORK

An early proposal for volume previews in a database search is described in [HES85]. The "Dining out in Carlton" example was provided to illustrate a search technique based on the volume preview of the number of the available restaurants. However, query previews were not exploited to support dynamic queries and querying in networked information systems.

Retrieval by reformulation is a method that supports incremental query formation by building on query results. Rabbit [Wil84] and Helgon [FNL89] are examples of retrieval systems based on the retrieval by reformulation paradigm, which is also the basis of the two-phase query formulation approach.

Harvest [BDH94] provides an integrated set of customizable tools for gathering information from diverse repositories, building topic-specific and searching them. Harvest could be used to maintain and update the metadata servers where users can extract information and store it locally in order support dynamic queries in both the query preview and refinement phases. However, Harvest, just like other WWW browsers, still applies the traditional querying technique based on keywords. In order to express a complex query, a more visual query interface may be effective.

Visualization techniques are increasingly being used to show results. In INQUERY [VN95], a ranked output information retrieval system for library catalog, the interface illustrates how the query results are related to the query words, helping users to reformulate the query. The Butterfly [MRC95] creates a virtual environment that grows under user control as asynchronous query processes link bibliographic records to form citation graphs.

CONCLUSIONS

The prototype dynamic query preview interface was presented to subjects as part of a Prototyping Workshop organized by Hughes Applied Information Systems. The prototype was evaluated by a dozen of NASA Earth Scientists who use EOSDIS to extract data for their research.

This was part of a larger evaluation effort and the evaluators reviewed several other querying interfaces during the day. In this evaluation, subjects received no training about the prototype, and were given five tasks. Subjects reacted positively to the new concepts in the query preview and query refinement interfaces. They agreed that the visual feedback provided in the query preview interface allows the user to pick data intuitively. Subjects also expressed their satisfaction with the visual feedback.

Formulating queries using query previews has many potential advantages:

- eliminates zero-hit queries
- reduces network activity and browsing effort by preventing the retrieval of undesired datasets
- represents statistical information of the database visually to aid comprehension and exploration
- supports dynamic queries, which aids users to discover dataset patterns and exceptions
- suitable to novice, exploratory, or expert users

Volume preview tables can become rather large if combinations are to be previewed accurately or if large numbers of previewing attributes or attribute values are chosen. But the benefits of the query preview technique is that it remains always possible the reduce the number of attributes or the granularity of the selections so that query preview is possible, allowing users to reduce the scope of the query in an informed and rapid way. The size of the preview table can also be adapted to users' work environment (network speed, workstation type) or preferences.

ACKNOWLEDGEMENTS

This work is supported in part by NASA (NAG 52895 and NAGW 2777) and NSF (EEC 94-02384 and IRI 96-15534).

REFERENCES

AS94 C. Ahlberg and B. Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Proc. of the ACM CHI94 Conf.,* 1994, pages 313-319.

BDH94 C. M. Bowman, P. B. Danzig, D. R. Hardy, U. Manber, and M. F. Schwartz. The Harvest information discovery and access system. In *Proc. of the Second International Conf. on the World Wide Web*, 1994, pages 763-771.

DPS96 K. Doan, C. Plaisant, and B. Shneiderman. Query previews in networked information systems. In *Proc. of the Forum on Advances in Digital Libraries.* IEEE Computer Society Press, 1996, pages 120-129.

DPS97 K. Doan, C. Plaisant, B. Shneiderman, and T. Bruns. Interface and data architecture for query previews in networked information systems. University of Maryland Department of Computer Science Technical Report (submitted for publication), 1997.

FNL89 G. Fischer and H. Nieper-Lemke. HELGON: Extending the retrieval by reformulation paradigm. In *Proc. of ACM CHI'89 Conf.* , 1989*,* pages 333-352.

HES85 D. L. Heppe, W. H. Edmondson, and R. Spence. Helping both the novice and advanced user in menu-driven information retrieval systems. In *Proc. of British HCI85 Conf.*, 1985*.,* pages 92-101.

MRC95 J. D. Mackinlay, R. Rao, and S. K. Card. An organic user interface for searching citation Links. In *Proc. of the ACM CHI95 Conf.,* 1995, pages 67-75.

Shn94 B. Shneiderman. Dynamic queries for visual information seeking. *IEEE Software 11*, 6, 1994, pages 70-77.

Tan96 E. Tanin, R. Beigel, and B. Shneiderman, Incremental Data structures and algorithms for dynamic query interfaces. *ACM SIGMOD Record 25*, 4, Dec. 96, pages 21-14

VN95 A. Veerasamy and S. Navathe. Querying, navigating and visualizing a digital library catalog. (URL http://www.csdl.tamu.edu/DL95/) In *Proc. of the Second International Conf. on the Theory and Practice of Digital Libraries*, 1995.

Wil84 M. D. Williams. What makes RABBIT run? In *International Journal of Man-Machine Studies 21*, 1984, pages 333-335.

Will93 C. Williamson and B. Shneiderman. The dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system, *Proc. ACM SIGIR `92 Conference*, ACM, New York, NY, 1992, pages 338-346