CAR-TR-788
CS-TR-3524
ISR-TR-95-90 Oct. 1995

QUERY PREVIEWS IN NETWORKED INFORMATION SYSTEMS

Khoa Doan, Catherine Plaisant and Ben Shneiderman†
Human-Computer Interaction Laboratory &
Department of Computer Science†,
Institute for Systems Research†
University of Maryland
College Park, MD20742
e-mail: {doan,plaisant,ben}@cs.umd.edu
phone: (301) 405 2725 ; fax: (301) 405 6707

ABSTRACT
In a networked information system, there are three major obstacles facing users in a querying process: *network performance, data volume* and *data complexity*. In order to overcome these obstacles, we propose a two-phase approach to dynamic query formulation by volume preview. The two phases are the *Query Preview* and *Query Refinement*. In the *Query Preview* phase, users formulate an initial query by selecting desired attribute values. The volume of matching data sets is shown graphically on *preview bars* which aid users to rapidly eliminate undesired data sets, and focus on a manageable number of relevant data sets. Query previews also prevent wasted steps by eliminating zero-hit queries. When the estimated number of data sets is low enough, the initial query is submitted to the network, which returns the metadata of the data sets for further refinement in the *Query Refinement* phase. The two-phase approach to query formulation overcomes slow network performance, and reduces the data volume and data complexity problems. This approach is especially appropriate for users who prefer the exploratory method to discover data patterns and exceptions during the query formulation process. Using this approach, we have developed dynamic query user interfaces to allow users to formulate their queries across a networked environment.

Keywords: Direct Manipulation, Dynamic Query, Information System, Network, Preview Bar, Query Preview, Science Data, Volume Preview, User Interface.

INTRODUCTION

Dynamic Queries

For the past several years, research at the Human-Computer Interaction Laboratory has focused on creating dynamic query user interfaces that apply the principles of direct manipulation to the database environment:

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

Dynamic queries involve the interactive control by a user of visual query parameters that generate a rapid (100 ms update), animated, visual display of database search results. The dynamic query approach lets users rapidly, safely, and even playfully explore a database. They can quickly discover where there are clusters, exceptions, gaps, or outliers, and what trends ordinal data reveal [Shn94]. An experiment was conducted comparing the dynamic query interface with a form-based interface and a natural language interface [WS92]. This experiment demonstrated the strengths of dynamic queries for complex queries, trend analysis and exceptions.

Information Retrieval

Exploration of large networked information resources becomes increasingly difficult as the volume grows. There are three major problems :

- **Network Performance:** Current network technology does not support rapid extraction of data, therefore retrieving large amounts of information on the network is often considerably slow, especially when the network traffic is high.
- **Data Volume:** The total amount of data sets, which are available from many different data servers on the network, is large. The data volume, which can reach hundred thousands or millions of data points, is too large for users to make decisions on which portions they want to extract.
- **Data Complexity:** The large number of attributes of a data set poses a challenge to users as it can be difficult to remember the type and name of all data set attributes, and is even harder to understand the relationship between the data set attributes during the querying process.

Traditionally, there are two strategies for information seekers to quickly and efficiently obtain the data in large information retrieval systems [Mar95]. *Analytical* strategies depend on careful planning, the recall of query terms, relevant iterative query formulation and examination of results. *Browsing* strategies are heuristic and opportunistic and depend on recognizing relevant information. The *analytical* strategies require users to have an intensive knowledge of application domains, and to be skillful in reasoning. The *browsing* strategies are difficult to use when the data volume is extremely large. Our information seeking strategies apply dynamic queries in a two-phase approach to query formulation by volume preview to combine the advantages of the *analytical* and *browsing* strategies. For example, dynamic queries are used to reduce the cognitive loads required in the *analytical* strategies. A two-phase approach reduces the number of undesired data sets and focuses on a manageable number of relevant data sets, which overcomes slow network performance, data volume and data complexity problems in the *browsing* strategies.

In the next section, we present a short survey of related work. Next, we demonstrate how we perform queries by a sequence of volume previews in a simple application called the Restaurant Finder. We then introduce new concepts and foundations for the two-phase approach to query formulation by volume preview, and describe a dynamic query user interface to the EOSDIS (Earth Observing System – Data and Information Systems), which assists users to formulate their queries in a very large networked information system. Finally, conclusions and future work are presented.

RELATED WORK

At present, extracting information from the network is performed using World-Wide-Web browsers such as Netscape or Mosaic. The querying technique is primarily based on keywords. Using this traditional technique, users may specify how many data a query should return (e.g. 20) but they never can estimate how many data were ignored, and how representative all the available data are. Querying is time consuming for it frequently retrieves undesired data, or gets zero-hit queries. Users also often fail to find the data if keywords cannot be guessed/found. This technique also suffers from slow network performance.

The Butterfly system was developed for simultaneously exploring multiple DIALOG bibliographic databases across the Internet using 3D interactive animation techniques [MRC95]. The key technique used by Butterfly is to create a virtual environment that grows under user control as asynchronous query processes link bibliographic records to form citation graphs. Asynchronous query processes reduce the overhead associated with accessing networked databases, and automatically formulated link-generating queries reduce the number of queries that must be formulated by the user. However, the authors confirm that Butterfly is hard to use without the support of a visual language after the experiment [MRC95].

The Attribute Explorer is a graphical interactive tool for visualizing the relationships within multi-attribute data sets [TSWB94]. In the Attribute Explorer, each attribute is mapped to a single dimensional representation (interactive histograms). Sections of an attribute's histogram can be selected by a variety of means (e.g. buttons, sliders, etc). The effect of one attribute on the others can be explored by selecting values of interest, and viewing the changes in the histograms. The Attribute Explorer is useful for perceiving trends and outliers in the multi-attribute data sets. However, there is neither discussions of applying the technique for querying data in a networked environment nor on how to handle complex data sets.

The Aggregate Manipulator (AM) allows users to create and decompose aggregates, which are groupings of data, and see their derived properties. In [GR94], a combination of the Aggregate Manipulator and Dynamic Query provide a highly useful tool for data manipulation functionalities in exploring large data sets such as: controlling scope, selecting focus of attention, and choosing level of detail. The method has been used to implement a data exploration interface to a large real-estate application. However, the system doesn't deal with querying data in a networked environment.

THE RESTAURANT FINDER

The Restaurant Finder is designed to help users identify restaurants that fulfill their desires. Users first specify a few criteria of the restaurants that they want, reduce incrementally the number of the available restaurants to a manageable size, and then submit their requests to the network to retrieve further information, and continue refining their queries with additional criteria.

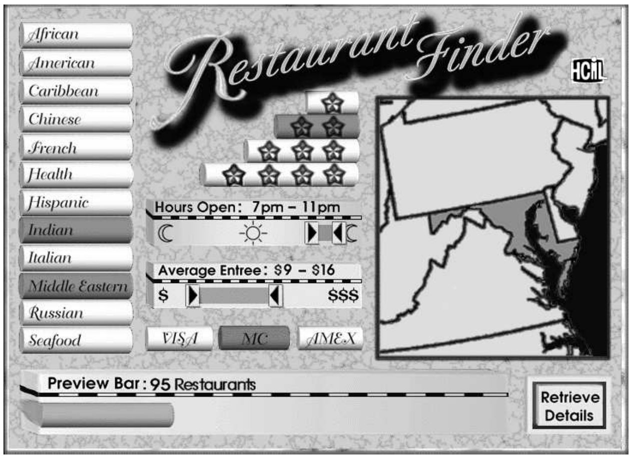


Figure 1: Display of the Restaurant Finder Preview Panel.

Initially, there are approximately 50,000 restaurants available for selection in the North East area. The Restaurant Finder aids users to reduce the number of the selected restaurants to under 100, so that users can retrieve more detailed information from the network. The Restaurant Finder's user interface provides sliders and buttons for selecting the relevant cuisine, range of cost, range of hours, geographic regions, rating, and charge cards (see figure 1). As selections are made, the preview bar on the bottom displays the volume of the restaurants in the database that satisfy the users' request. The preview bar allows users to explore safely through the database, and eliminates the chance of requesting information that is not available. To allow volume preview updates within a tenth of a second, the attribute values must be kept in the high speed storage. Users can quickly see if there are any Chinese restaurants open after midnight. Users may discover that there are more Chinese restaurants than Italian restaurants, but more Italian restaurants than Chinese restaurants, that are open after midnight.

When the size of the volume preview bar is below the recommended level, users can click on the retrieve button. Detailed information is retrieved from the network. The map then becomes local, showing each restaurant as a dot, and more parameters become available, for instance, parking space, number of tables, meeting rooms, disabled access, etc. The query can then be further refined by selecting more precise values. Details on demand for each restaurant remaining after the query refinement (e.g. the full menus, reviews, photographs, etc) can be obtained from the network.

TWO-PHASE QUERY FORMULATION PROCESS

Our approach is based on the volume preview table, which is used to update preview bars during the Query Preview Phase and the Query Refinement Phase. The goal of this approach is to reduce the volume of the data sets to a manageable size, and to prevent zero-hit queries before submitting queries to the network. The reduction process is performed incrementally by selecting rough ranges of values of a few attributes in the Query Preview Phase to selecting more precise values or exact values of more attributes in the Query Refinement Phase (figure 2). The reduction of the number of the available data sets in the second phase gives users more control over the attribute values. Only a few attributes are displayed in the Query Preview Panel but in the Query Refinement Panel, an exhaustive list of all common attributes are presented for further selection. A complete list of the attributes of any data set can also be obtained for details on demand.

	QUERY PREVIEW	QUERY REFINEMENT
Number of Data Sets	Very large	Manageable (each one is selectable for details-on-demand)
Number of Attributes	Limited	Exhaustive List of Common Attributes
Selection of Attr ibute Values	Rough Ranges	More Precise or Exact Values

Figure 2: A comparison table of the two phases of the query formulation process.

The architecture of the two-phase approach to query formulation is illustrated in figure 4. In these two phases, the query is created by direct manipulations such as adjusting the sliders, pressing buttons or using the pointing devices (e.g. a mouse or a trackball).

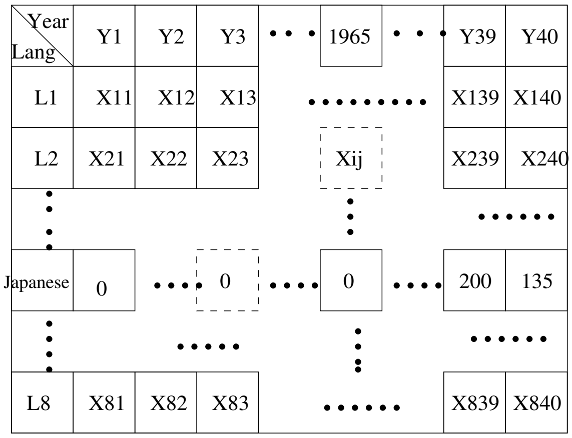
This approach depends on the network data centers being willing to produce and publish tables of contents for relevant sets of categories. Alternatively, web browsers could extract this information. The tables of content should be small enough to load into the high speed storage to support dynamic queries.

The idea of using table of contents in the Query Preview Panel is analogous to using the table of contents/index while looking for the relevant information in a new book. Using the table of contents/index helps to estimate the types and size of the available data without reading the book. In a networked information system, the volume preview table is produced by intersection on multiple tables of content.

For example, a data center might have N documents (in the millions), and two tables of contents with cardinalities n1 and n2, for example 40 years and 8 languages. The volume preview table would have n1 * n2 values (320 entries for our example) and its size is independent of N. Such a volume preview table would enable users to discover that there were no Japanese papers before 1965 without even going to the data center (figure 3). The volume preview table has to be updated periodically (for example, daily). This is a limitation of query previews but the advantages are substantial if frequent queries are anticipated.

In the Query Preview Panel, the volume preview table is visualized using bars or a combination of bars, shaded maps, pie charts (figure 5), which are called the preview bars. A preview bar is used to display the estimated number of the data set hits for an attribute value (called *attribute preview bar*), or for the query (called *query preview bar* (figure 1)). In a preview bar (either an attribute preview bar or the query preview bar), the two colors, such as gray and white, may be used to indicate a selection or a non-selection. The width and length (or even the area) of the preview bars is proportional to the size of the volume it represents. All the preview bars are tightly-coupled in the Query Preview Panel [AS94]. When an attribute value or range is modified, all the preview bars are updated and immediately visible (see figures 6,7,8 and 9).

The query preview bar also has a recommended level which can be set by users in the Query Preview Phase. When the number of the data set hits exceeds the currently selected recommended level, the preview bar displays a message to



Xij: the number of the documents in language Li published in year Yj

Figure 3: The volume preview table produced by the two tables of contents with 40 years and 8 languages. This table is used to update the preview bars

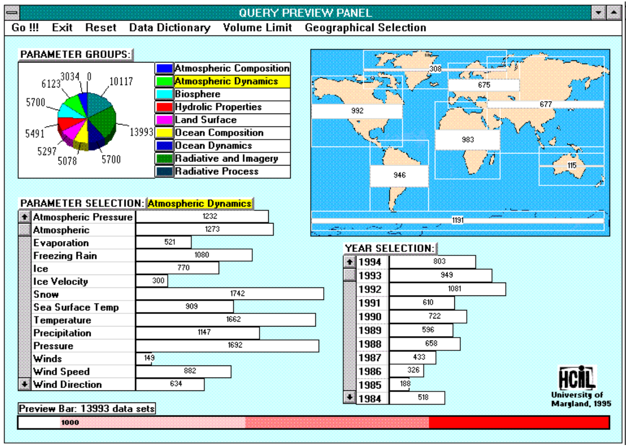


Figure 6: Display of the Query Preview Panel after selecting a parameter group: `Atmospheric Dynamics`.

users warning them that the number of hits will result in delays and slow operations in the Query Refinement Phase.

The volume preview table is especially a useful starting point for the query formulation process when users don't have an extensive knowledge about the data. In summary, the benefits of the volume preview in the Query Preview Phase are:

- Provide users useful statistical information about the metadata of the data sets without having to retrieve the data sets from the network;
- Aid users to eliminate rapidly undesired data sets, and guide them to focus on the data sets of interest and of a manageable size;
- Prevent retrieving undesired data sets via the network which is costly and time-consuming;
- Eliminate the chance of retrieving with zero hits;
- Support dynamic queries; and
- Aid users to discover data set patterns and exceptions.

DYNAMIC QUERY USER INTERFACES TO THE EOSDIS

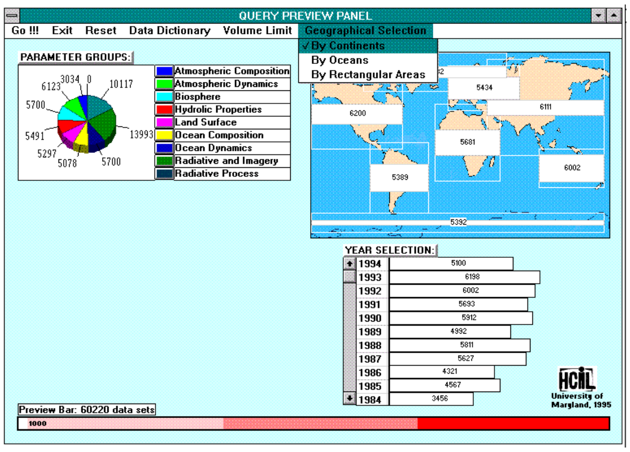


Figure 5: Initial Display of the Query Preview Panel.

The Earth Observing System (EOS) Data and Information System (EOSDIS) is a comprehensive data and information system, developed by NASA under the Mission to Planet Earth (MTPE) Program. EOSDIS will manage data from NASA's past and current Earth science research satellites and field measurement programs, providing data archive, distribution, and information management services. Currently, the V0 IMS (Information Management System) is the sole user interface that provides access to the EOSDIS so that EOS scientists and users can use it to search and study the EOS data [Doa95]. The V0 IMS is difficult for EOSDIS users without a specific knowledge of the science data to find the right data sets due to the extremely large volume of the available data sets in the EOSDIS. In the VO IMS, users may specify how many data sets a query should return (e.g. 20) but they never can estimate how many data sets were ignored, and how representative all the available data sets are.

In addressing the limitations of the V0 IMS, we present a Dynamic Query User Interface to the EOSDIS consisting of the *Query Preview Panel* and *Query Refinement Panel* as illustrated in figure 5 and 10 respectively.

The Query Preview Panel

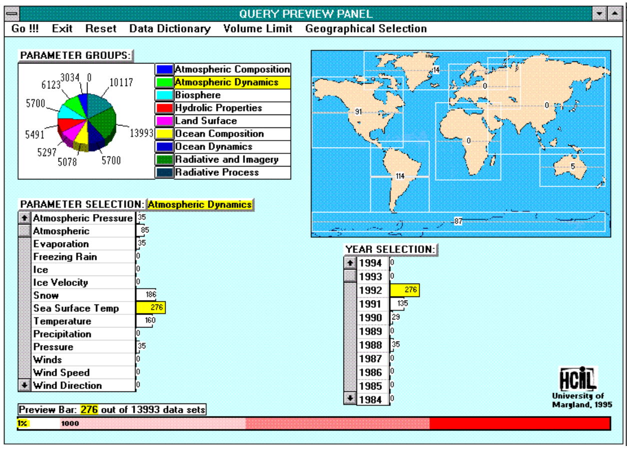


Figure 8: Display of the Query Preview Panel after selecting a specific year: `1992`.

A Visual Basic prototype of the Query Preview Panel is described in this section. There are three selected attributes displayed in the interface which are the *parameter*, *spatial*

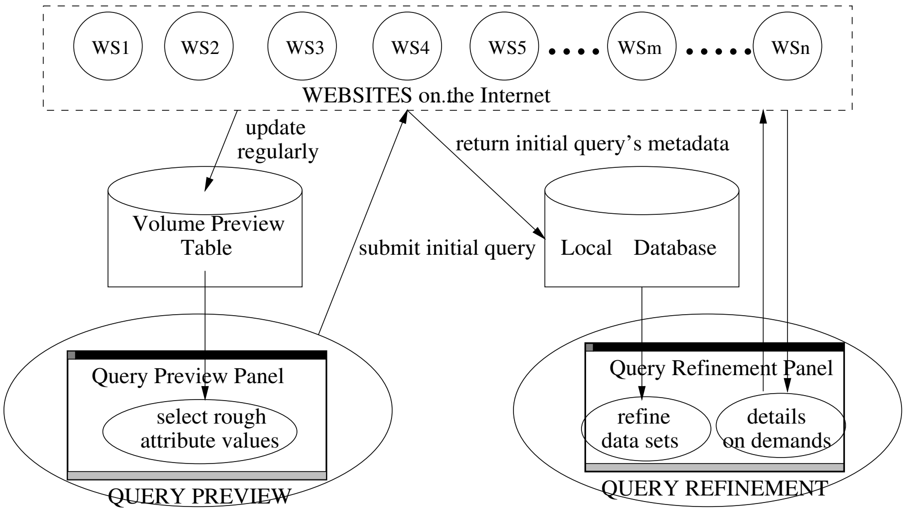


Figure 4: Architecture of Two-phase Dynamic Query Approach for Networked Information Systems.

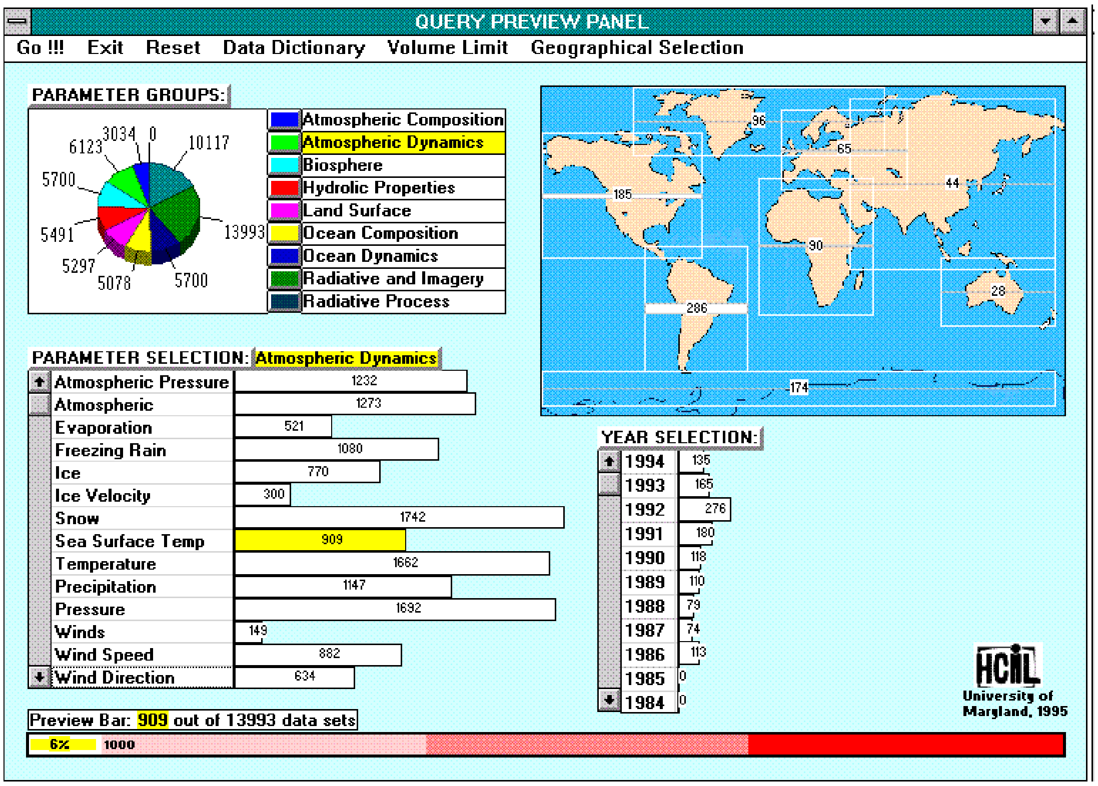


Figure 7: Display of the Query Preview Panel after selecting a parameter value: `Sea Surface Temp`.

and *temporal coverage*.

The parameters of EOS data sets are classified into 9 groups in terms of the types of the data sets they represent (e.g. Atmospheric Composition, Atmospheric Dynamics, etc). The spatial coverage are defined by the continents (e.g. Africa, Asia, etc), oceans (e.g. Pacific, Atlantic, etc) or a selectable grid map, and the temporal coverage is measured in terms of years (e.g. 1986, 1987, etc). When the Query Preview Panel starts off, it first displays the number of data sets for each parameter group, selected region and year respectively, in the form of the attribute preview bars, as shown in figure 5. The size of the data sets for each attribute value is proportional to either the area (e.g. the attribute preview bars of the continents), or to the length (e.g. the attribute preview bars of the years and parameters) of the corresponding rectangular bars. The query preview bar, which is on the bottom of the Query Preview Panel, displays graphically the total number of the selected data sets in the gray part on the left section of the bar, and the increasingly red parts on the right section of the bar represent the excessive region (above the recommended level, which is 1000 in figure 7). When the number of the selected data sets exceeds the recommended level (which results in the overlapping of the gray part over the red parts), a warning message is displayed to users.

An initial query in the Query Preview Panel may be formulated by first selecting the parameter group of interest, which results in the display of all the available parameters in that group and its corresponding preview bars. As a result of the parameter group selection, the attribute preview bars for each continent and year are also updated to display the corresponding number of data sets that contain one or more parameters of the selected parameter group.

For example, if users might be interested in the temperature of US Coastal Waters. Since the US Coast is in the `"North America"` area, they may select the *"By Continents"* option from the *Geographical Selection*'s popup menu. Using the Data Dictionary facility, users discover that the parameter `"Sea Surface Temp"` that is used to study the temperature of coastal waters is in both the `"Atmospheric Dynamics"` and `"Ocean Dynamic"` parameter groups. The pie chart of the parameter groups shows that there are more data sets in the `"Atmospheric Dynamics"` than in the `"Ocean Dynamic"`. Hence, users may select the `"Atmospheric Dynamics"` in order to get more data. The result of the parameter group selection is illustrated in figure 6, in which there are 13993 data sets in the query preview bar (the bottom rectangular bar in the Query Preview Panel). Users then select the parameter `"Sea Surface Temp"`, which results in the change of the attribute preview bars representing each continent and year (figure 7). The updated preview bars represent the number of data sets that contain the parameter `"Sea Surface Temp"`, and the total number of the selected data sets in the query preview bar is now reduced to 909 as illustrated in figure 7. Users then further reduce the number of the data set hits by choosing a specific year. The attribute preview bars of the non-selected attribute values disclose to users the total of the data set hits they might get if selected. For example, users wouldn't select the years `1984` or `1985` since the corresponding preview bars indicate that there are zero data sets in the year 1984 and 1985 (figure 7). It also reveals that there are the most data sets on the `"Sea Surface Temp"` in the year `1992` (hence, it was selected). The total number of the selected data sets is now reduced to 276, and users can continue to reduce the volume of the relevant data sets to 91 by selecting `"North America"` which contains the US Coast (figure 8). Finally, users submit the initial query to the DAACs (Data Acquisition Archive Centers) for the extraction of metadata of the selected data sets.

The Query Refinement Panel

The Query Refinement Panel supports dynamic queries over a local database that stores the metadata of the data sets extracted from the Query Preview Panel. The metadata contains the information of all the attributes of the data sets such as the parameter, sensor, platform, project, data archive centers, processing data level, time, location which are also visually represented in the interface (figure 10). The main function of the Query Refinement Panel is to support further refinement for the data sets in the first step. Each data set is now represented as a line in the starfield display [AS94], which is referenced by the two axes representing the size (vertical axis) and the time period (horizontal axis) of the data sets respectively. By randomly selecting the regions in the *"North America"* map, users may discover that there are more data sets in the US West Coast (hence it was selected). Users further refine the query by selecting more precise values for the parameter, sensor, platform, project, data archive centers, processing data level, etc. When the query is completely refined, users select the number of returned granules [1] per data set . Users may want to access to details on demand by clicking on a specific data set in the starfield display. The image of the granules and full details of the selected data set are retrieved from DAACs. Subsequently, these graphical and detailed information of the data set are displayed at the bottom right of the Query Refinement Panel, as shown in figure 11. Users can use the "timeline" slider to eliminate the data sets of undesired periods from the starfield display.

In both the Query Preview Panel and Query Refinement Panel, the system also supports multiple selection of the attribute values and going back and forth between the two phases. However, the system has several limitations. The attribute preview bar only gives the conjunction of currently selected data sets. It is rather time consuming to go back and forth between the two phases due to slow network performance. Our solution to network transfer problems in some sense defeats the power of relevance feedback and query reformulation but this will need to be tested.

CONCLUSIONS AND FUTURE WORK

The two-phase approach to query formulation by volume preview appears to be an efficient method to extract or query a very large and complex database. This approach also demonstrates how dynamic queries can be used in a networked environment via the development of a user interface to the EOSDIS. Future work includes:

- Extend the Query Preview Panel with capabilities to provide users an option to select the attributes of their choice, and to support both generic and specific and user-defined visualization schema (for the selected attributes) for presentation.
- Integrate the Query Preview Panel into multiple platforms (eg. Unix, Windows) and information systems (e.g. NASA's V0 IMS).
- Research on data structures and algorithms to support rapid multi-dimensional search in a large or very large database.
- Conduct a comparative query performance experiment between the traditional query approach and the approach described here in the context of a networked query environment,

[1] In the EOSDIS, a data set may consist of thousands or millions granules which is a smallest unit that can managed and displayed in the interface. Hence, only a modest number of granules are retrieved for browsing

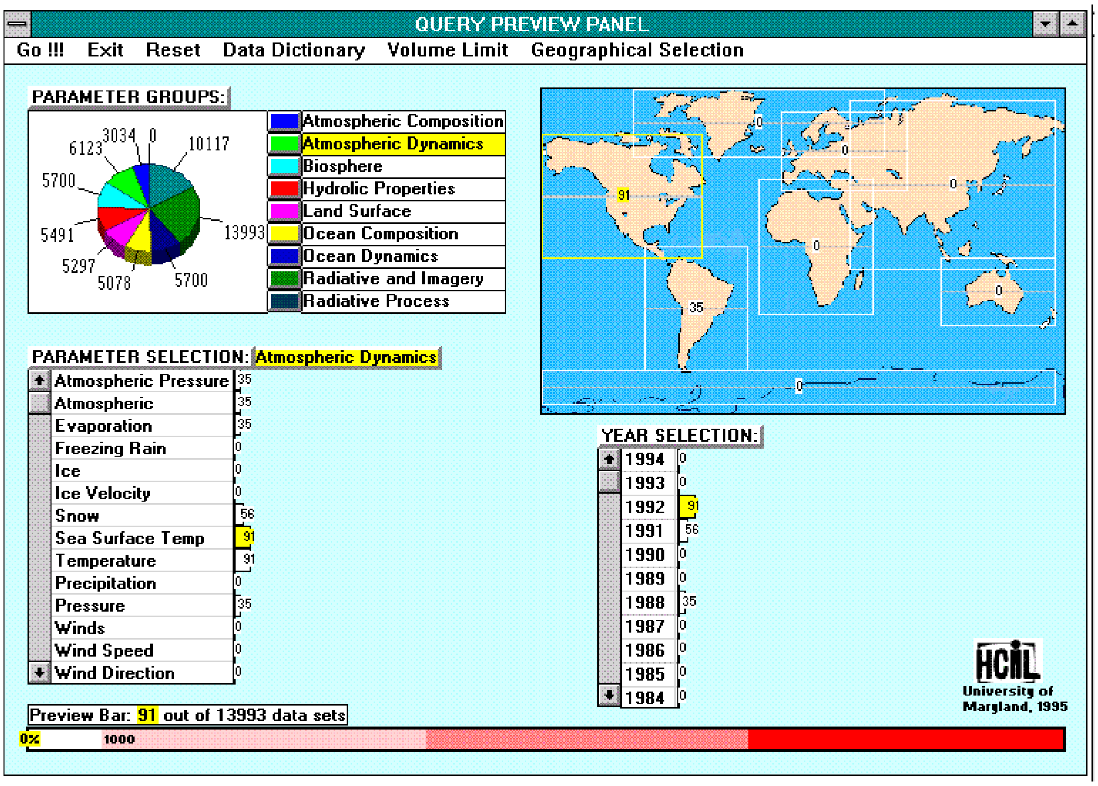


Figure 9: Display of the Query Preview Panel after selecting `North America`.

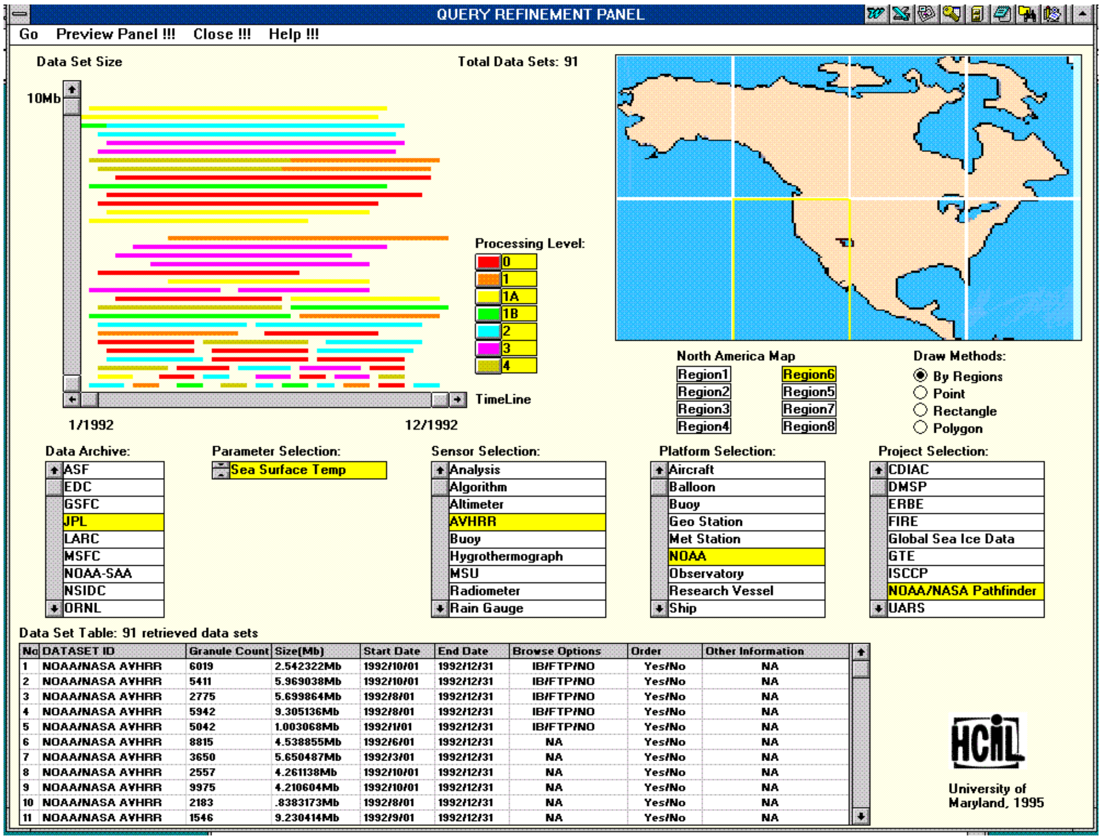


Figure 10: Display of the Query Refinement Panel in the Data Set Refinement Step

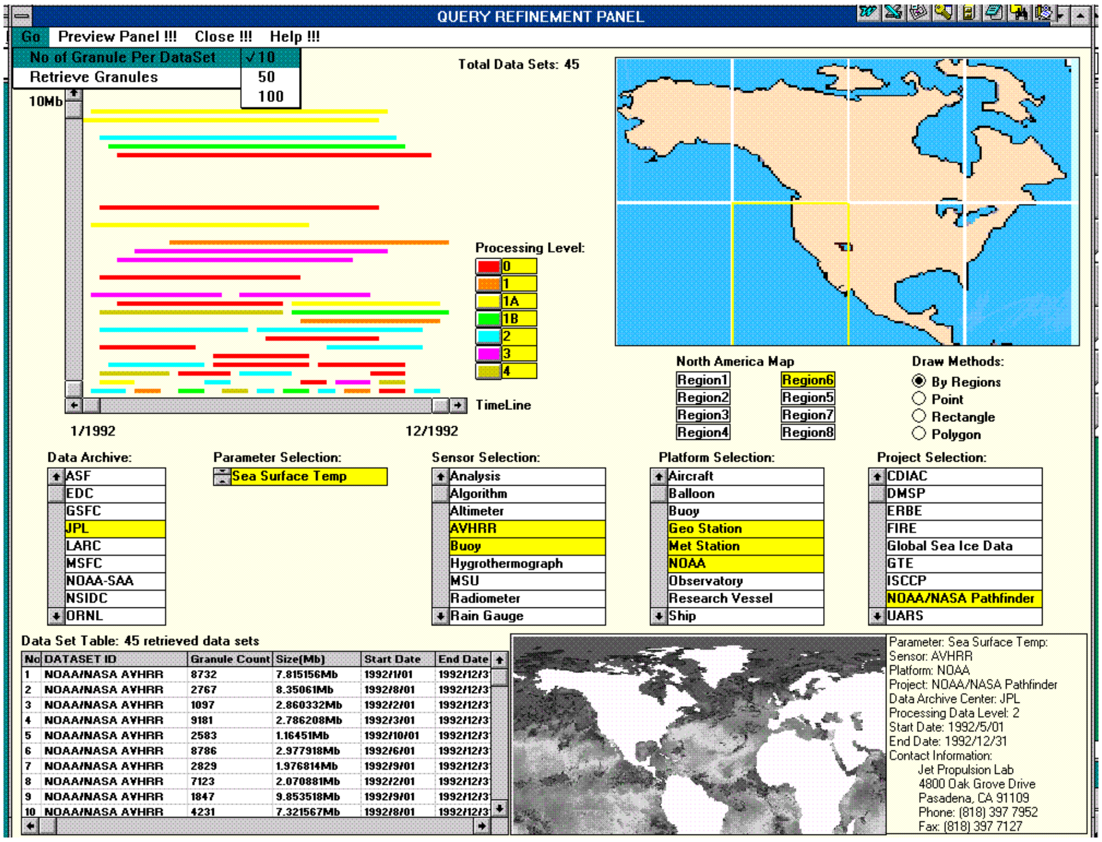


Figure 11: Display of the Query Refinement Panel in the Details-on-Demand Step.

with different categories of users, query tasks and application domains.

ACKNOWLEDGEMENTS

This work is supported in part by NASA (NAG 52895) and by the NSF grant NSF EEC 94-02384. We thank Teresa Cronnell for her graphic design of the Restaurant Finder prototype. Our thanks also go to Gary Marchionini, Robin Pfister, and Chris Rouff for reviewing the draft paper.

REFERENCES

AS94. C. Ahlberg and B. Shneiderman. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In *The Proc. Human Factors in Computing Systems Conf*, pages 313–319, 1994.

Doa95. Khoa Doan. Initial Review of the V0 IMS GUI. Technical report, HCIL, Unisersity of Maryland, 1995.

GR94. J. Goldstein and S. F. Roth. Using Aggregation and Dynamic Queries for Exploring Large Data Sets. In *The Proc. Human Factors in Computing Systems Conf*, pages 23–29, 1994.

Mar95. G. Marchionini. *Information Seeking in Electronic Environments*. Cambridge Series on HCI, 1995.

MRC95. J. D. Mackinlay, R. Rao, and S. K. Card. An Organic User Interface for Searching Citation Links. In *The Proc. Human Factors in Computing Systems Conf*, pages 67–75, 1995.

Shn94. B. Shneiderman. Dynamic Queries for Visual Information Seeking. *IEEE Software*, pages 70–77, 1994.

TSWB94. L. Tweedie, B. Spence, D. Williams, and R. Bhogal. The Attribute Explorer. In *The Video Proc. Human Factors in Computing Systems Conf*, pages 435–436, 1994.

WS92. C. Williamson and B. Shneiderman. The Dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system. In *Proc. ACM SIGIR Conf.*, pages 339–346, 1992.