

# Dynamic Queries: Database Searching by Direct Manipulation

*Ben Shneiderman, Christopher Williamson, and Christopher Ahlberg*

Computer Science Department and Center for Automation Research  
Human-Computer Interaction Laboratory  
University of Maryland, College Park, MD 20742  
ben@cs.umd.edu

## INTRODUCTION

This video explores the application of direct manipulation to information exploration. Specifically, it introduces the idea of a dynamic query, which empowers the user to search a database of information in the task domain while requiring minimal syntax or computer knowledge. Key features that separate a dynamic query from current information retrieval systems are its direct manipulative nature, providing immediate search feedback with object interaction, and a harmonious display of the query and results in a graphical environment appropriate for the task domain [3].

A slider serves as a metaphor for the operation of entering a value for a field in the query - it provides a mental model [2] of the range. Changing the value is done by a physical action - sliding the drag box with a mouse - instead of entering the value by keyboard. By being able to slide the drag box back and forth and getting immediate updates of the query results, it is possible to do dozens of queries in just a few seconds. The operation is incremental and if the query result is not what users expected the operation is reversible by simply sliding the slider back. Error messages are not needed - there is no such thing as an 'illegal' operation or incorrect syntax.

The results of the query are displayed in a graphical format near the sliders. The interaction between the visualization and the query mechanism is important. The sliders have to be placed close to the visualization to reduce eye movement. The highlighting of elements should be in harmony with the coloring scheme of the slider. The color of the area to the left of the drag box on the slider bar is the same as the highlighted elements in the visualization, because the values to the left of the drag box are the values that satisfy the query.

## APPLICATION: PERIODIC TABLE

The first application of dynamic queries explored was an educational program for the periodic table of elements. Designed by Christopher Ahlberg on the Sun SparcStation 1+, it allows users to set properties such as atomic number, atomic mass, electronegativity, etc. to highlight a certain number of elements in the periodic table (Figure 1). This lets users explore

how these properties interact with each other. Other interesting discoveries can be made regarding trends of properties in the periodic table - such as how electronegativity increases from the lower left corner to the upper right corner of the periodic table and to find exceptions to trends.

A user study was conducted on this interface and subjects were found to perform statistically significantly faster with the dynamic queries interface than with the form fill-in interface [1]. The subjects also showed a statistically significant preference for the dynamic queries interface over the other two interfaces.

This application also demonstrates the power of dynamic queries to find trends and exceptions in a database. The causality, incrementability, and the display's graphical nature make exceptions more apparent. Since most users have excellent visual discrimination and many people look for exceptions in a database (such as finding the cheap home in a nice neighborhood, or the weak part in an automobile), dynamic queries can help the user find them.

Finally, this application demonstrates a form of Query-by-Example [4], we call Query-by-Instance, whereby the user may select an item in the graphical display and initialize the slider's settings for those properties of that element. This can be very helpful for those who are not familiar with the values of the properties, but can specify that they wish to find elements with similar properties to that of a known element. The user can then manipulate the sliders to widen the range of specific properties from those of the specified instance.

## APPLICATION: DYNAMIC HOMEFINDER

The second application of dynamic queries explored was a practical public-access program for helping users find a Williamson on the IBM PC utilizing a touchscreen, it allows users to set properties with sliders such as proximity to work or school, number of bedrooms, cost, and services such as fireplaces and air conditioning (Figure 2). A map of the area is displayed beside these sliders with points of light designating individual homes that meet the properties specified. More specific information on a home could then be retrieved by selecting that home on the touchscreen.

This application shows that a relatively complex searching specification for a home can be designed using dynamic queries such that it is suitable for first-time users and public-access systems. Further, the proximity specification allows the user to

specify geographically the area and distance in which they are looking for a home. This simple area specification is rarely offered on current real estate searching systems. The trends, although less apparent than in the periodic table application, are still visible such as cost and type of homes in given neighborhoods. One of the classic problems with information retrieval is narrowing the results of the search. With typical command-line interfaces, it is too easy for a user to specify a query which fails to find anything. They modify their previous query, this time getting back too much information - the user is forced to guess values to narrow their results. Dynamic queries help to

overcome this all-or-nothing problem: users can easily fine-tune their search based on previous results. If the desired results fail to turn up enough hits, or too many, the user can simply fine-tune the sliders on one or more attributes until the desired number of hits is achieved.

This application dramatically demonstrates the feeling of causality conveyed to the user. The direct manipulative qualities of incrementability, reversibility, and smooth graphical feedback encourage the user to explore the database, freely manipulating the properties without fear of syntax errors or of getting lost.

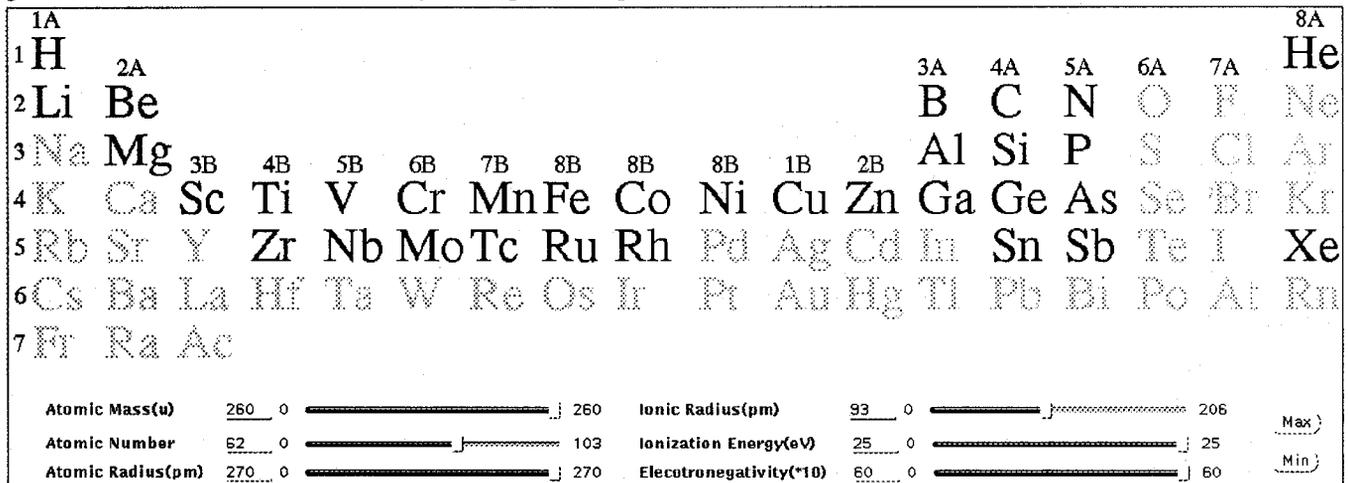


Figure 1. Application: Periodic Table of Elements

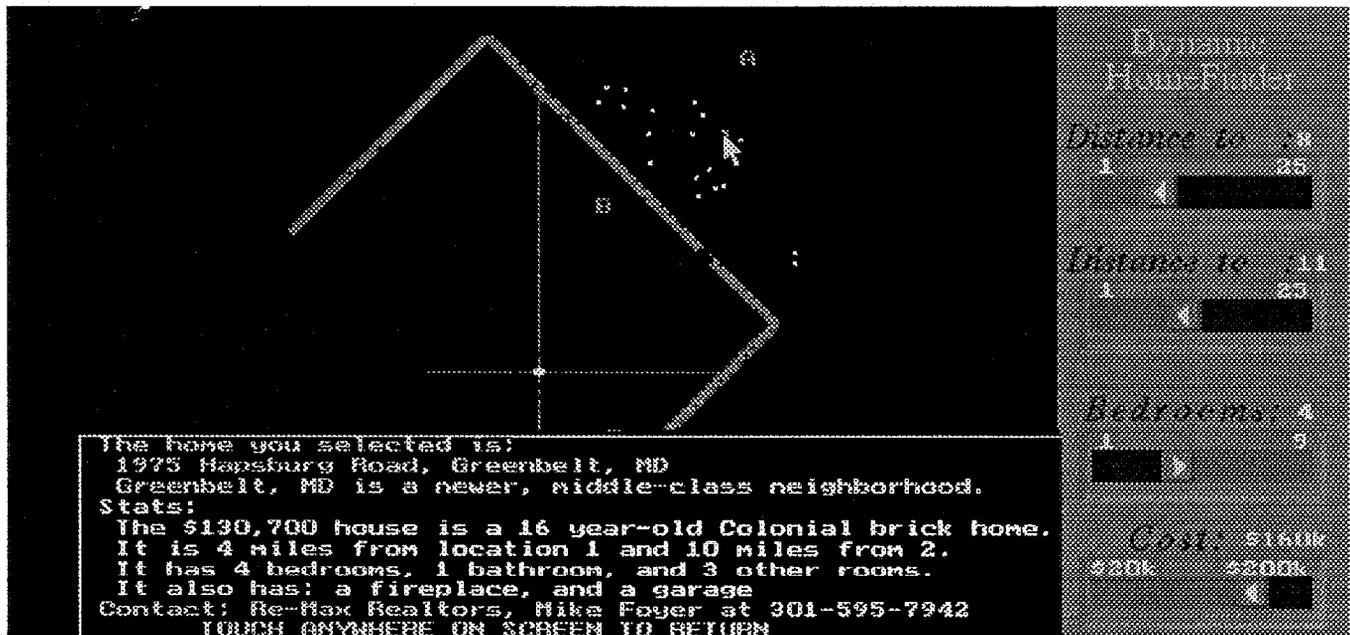


Figure 2. Application: Close-up of Dynamic HomeFinder

## REFERENCES

- Ahlberg, C., Williamson, C., and Shneiderman, B. Dynamic Queries for Information Exploration: An Implementation and Evaluation. *Proc. ACM CHI '92*.
- Shneiderman, Ben, Direct Manipulation: A step beyond programming languages. *IEEE Computer*, 16, 8 (August 1983), pp. 57-69.
- Williamson, C. and Shneiderman, B. *The Dynamic HomeFinder: Evaluating Dynamic Queries in a Real-Estate Information Exploration System*. CS-TR-2819, College Park, MD, 1991.
- Zloof M. Query-by-Example. National Computer Conference, AFIPS Press, 1975, pp. 431-437.