Zooming and Tunneling in Tioga: Supporting Navigation in Multidimensional Space

Allison Woodruff*, Peter Wisnovsky†, Cimarron Taylor†, Michael Stonebraker*, Caroline Paxson*, Jolly Chen* and Alexander Aiken*

* Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94720

† Illustra Information Technologies, Inc.
1111 Broadway, Suite 2000
Oakland, CA 94607

Abstract

In [7] we proposed a visual programming system called Tioga. The Tioga system applies a boxes and arrows programming notation to allow nonexpert users to graphically construct database applications. Users connect database procedures using a dataflow model. Browsers are used to visualize the resulting data.

This paper describes extensions to the Tioga browser protocol. These extensions allow sophisticated, flight-simulator navigation through a multidimensional data space. This design also incorporates wormholes to allow tunneling between different multidimensional spaces. Wormholes are shown to be substantial generalizations of hyperlinks in a hypertext system.

These powerful mechanisms for relating data provide users with great flexibility. For example, users can create magnifying glasses that provide an enhanced view of the underlying data.

1. Introduction

In [7], we presented Tioga, a new paradigm for user interaction with a database management system (DBMS). Tioga is motivated by the needs of scientific DBMS users in the SEQUOIA 2000 project [6]. Tioga uses the **boxes and arrows** notation popularized by scientific visualization systems such as AVS [8], Data Explorer [2], and Khoros [4]. Tioga improves upon these systems by providing sophisticated data management using the POSTGRES DBMS [5]. In the Tioga programming model, boxes represent user-defined database queries or browsers, and edges between boxes represent flow of data. This paradigm allows nonexperts to build visual programs called **recipes** by interactively connecting boxes together using a graphical user interface.

Browsers are attached to recipes wherever data needs to be visualized. These boxes provide a flight simulator interface with which users may view their data results. Navigation in this data space can become extremely complex and demands special support. This paper focuses on new constructs to support navigation in multidimensional space.

2. Showing enhanced detail

To eliminate clutter in the display and to orient the user, data should have different representations when seen from different distances in multidimensional space. Intuitively, we wish to extend Tioga with the possibility of **zooming** into data to display more detail about screen objects. For example, the Kodak photoCD representation for digital images supports a 2K by 3K by 8-bit color image format, and in addition provides four other images of lesser resolution culminating in a 128 by 192 by 8-bit abstract [1]. A user would like the ability to see abstracts on the screen and then zoom in to view the images at a higher resolution.

To support zooming, we are extending the existing Tioga browsing protocol. In our design, data objects can have different abstracts that are produced by different recipes. An elevation map relates these recipes to each other by specifying which recipes are valid at what distance (elevation) from the object. An elevation map is used to control the invocation of different recipes as the user zooms in and out through the data space. Therefore, if the user zooms into the elevation range of a different

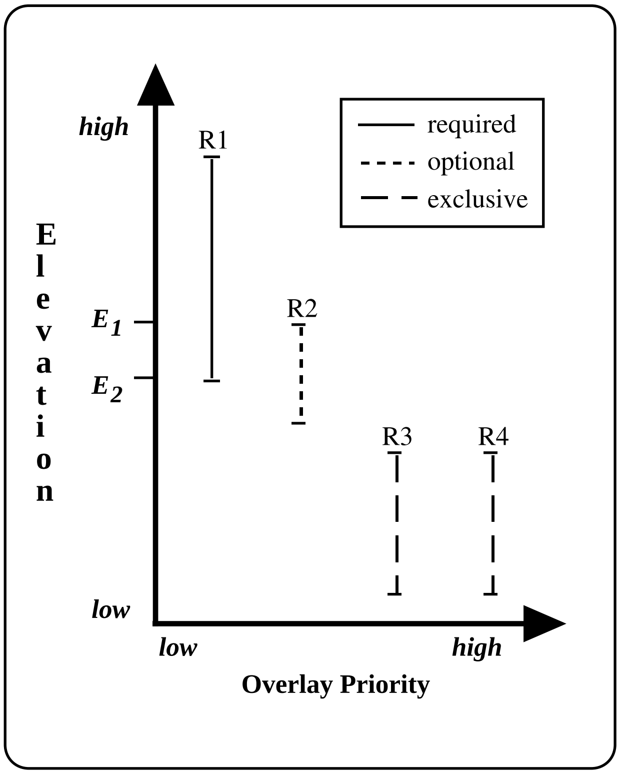

Figure 1
An Example Elevation Map

recipe, the recipe providing input to the browser is changed.

The elevation map also contains a collection of semantic restrictions on the display of recipe output. Each recipe in an elevation map can be required, optional, or exclusive.

Figure 1 shows an elevation map containing four recipes, R1 (state outline), R2 (census tracts), R3 (rivers), and R4 (highways). Each of these is valid in different elevation ranges. First the user sees the outline of California. Upon zooming to E_1, he or she has the option to see also the census tracts in the state. At the next transition point, E_2, the outline of the state is no longer visible and the census tracts are optionally visible. Further zooming shows either the rivers or the highways of California, but not both.

Elevation maps allow a user to specify easily the semantics of the zoom operation, assuming that all recipes produce data in the same multidimensional space. We now turn to a mechanism for changing from one multidimensional space to another.

3. Changing multidimensional spaces

Enhanced detail implies a change in perspective within a multidimensional space. Users also want the ability to move to related multidimensional data spaces for new perspectives on the data, a functionality similar to that provided by hyperlinks [3]. For example, a user could zoom in on a map of Berkeley to find the Computer Science building. Additional levels of detail could yield documents corresponding to Computer Science technical reports. These documents should be displayed in a different context than the latitude/longitude coordinates appropriate for the map of Berkeley. When a document is being viewed, a further zoom could yield the image of the author or the layout geometry of his or her office. Again, a different multidimensional space should be used.

When a user changes to a new multidimensional space, objects have a spatial relationship that is unrelated to the relationship in effect on the near side. This behavior should be distinguished from a zoom operation where the same spatial relationship is present before and after the zoom. Therefore, we denote this operation of changing from one multidimensional space to another as **tunneling** through a **wormhole**, to differentiate it from zooming.

When the user tunnels through a wormhole, a new application is invoked taking in a run-time parameter, which is the identifier of the object or objects associated with the wormhole. The reader can readily observe that wormholes are a substantial generalization of hyperlinks in a hypertext system. A hyperlink is a wormhole in which there is a prespecified object or collection of objects on each side of the hole. Wormholes allow a run-time specified set of objects to be on each side.

4. Coordinating multiple browsers

We would like to allow the user to place one browser inside another. For example, the user could place a **magnifying glass** browser inside another browser displaying a map of California. This is analogous to reading a map of California with the aid of a physical magnifying glass.

Two browsers in the same recipe can be related in two different ways. First, they may be **independent.** In this case, when either browser moves in multidimensional space or zooms in elevation, then nothing automatically happens to the other browser. This behavior is appropriate when the browsers are displaying independent objects. Alternatively, one browser can be **slaved** to a **master** browser. In this case, whenever the user changes the master's position in multidimensional space, the slave's position automatically changes as well.

Independent or slaved browsers may be declared to share a window. Figure 2 shows two browsers, an outer browser and an inner browser. The inner browser displays a subset of the data from the outer browser. Each browser has separate controls that determine the content of its display. The outer browser uses the complete window to display its objects. The inner browser magnifies a portion of the data shown in the outer browser. The display of the inner browser is then overlaid on the display of the outer one, thereby creating a single composite display which can be rendered on the screen by the window manager.

The definitions presented above may be recursively extended so that an inner browser may in turn serve as an outer browser. This allows an arbitrary number of browsers to share a window, with a pairwise inner-outer relationship between them. Thus, a hierarchical collection of browsers can be defined. For example, a user may choose to have a magnifying glass on top of another magnifying glass, providing even more detail.

5. Conclusions

In this paper we have detailed mechanisms that allow navigation in multidimensional space. We first outlined a zoom capability that allows users to view data at different levels of detail. We next introduced the concept of wormholes. Tunneling through a wormhole allows users to view their data in the context of a new multidimensional viewing space. We then proposed mechanisms for coordinating browser behavior. Finally, we detailed the behavior of multiple browsers sharing a portion of the screen. In combination, these constructs allow users to create effects such as magnifying glasses. We are currently extending the existing Tioga system to incorporate the features discussed in this paper.

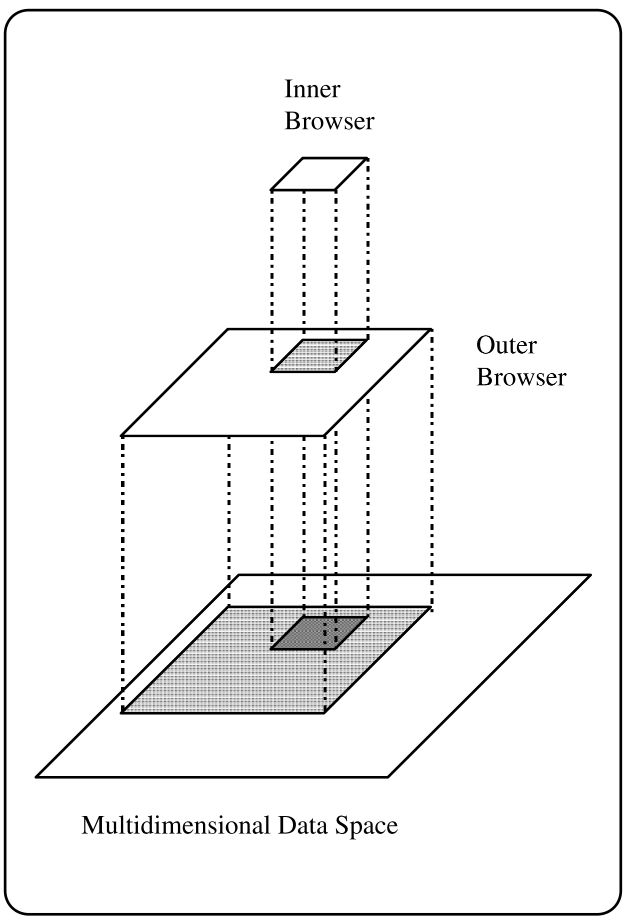

Figure 2
Viewing Regions of Nested Browsers

Acknowledgments

We would like to thank Nobuko Nathan, Jeff Sidell, Alan Su, and Jiang Wu for their contributions to this work in both discussion and implementation.

References

[1] Eastman Kodak Company, "Programmer's Guide for UNIX Systems," *Kodak PhotoCD Access Developer Toolkit*, 1992.

[2] Lucas, B., Abram, G., Collins, N., Epstein, D., et al., "An Architecture for a Scientific Visualization System," Proceedings of the 1992 IEEE Visualization Conference, Boston, MA, October 1992.

[3] Nielson, J. "The Art of Navigating through Hypertext," Communications of the ACM, March 1990.

[4] Rasure, J. and Young, M., "An Open Environment for Image Processing Software Development," Proceedings of the 1992 SPIE Symposium on Electronic Image Processing, February 1992.

[5] Stonebraker, M. and Kemnitz, G., "The POSTGRES Next-Generation Database Management System," Communications of the ACM, October 1991.

[6] Stonebraker, M. and Dozier, J., "SEQUOIA 2000: Large Capacity Object Servers to Support Global Change Research," SEQUOIA 2000 Technical Report 91/1, Electronics Research Lab, University of California, Berkeley, March 1992.

[7] Stonebraker, M., et al., "Tioga: Providing Data Management for Scientific Visualization Applications," Proceedings of the 1993 VLDB Conference, Dublin, Ireland, August 1993.

[8] Upson, C., Faulhaber, T., Kamins, D., Laidlaw, D., et al., "The Application Visualization System," IEEE Computer Graphics and Applications, July 1989.