Position Paper for CHI '95 Research Symposium (May 6-7, 1995, Denver, Colorado U.S.A.)

A Model for 3D Interaction with Hierarchical Information Spaces

Elisabeth Freeman and Susanne Hupfer
Department of Computer Science
Yale University
(freeman-elisabeth@cs.yale.edu, hupfer-susanne@cs.yale.edu)

Introduction

We consider a domain-independent model for 3D interaction with hierarchical information spaces. The model presents a virtual 3D workspace which allows the user to navigate a hierarchy in a controlled manner, with basic up and down movements. This restricted navigation is used to keep the user aware of her context in the information space: location, neighbors, and level. We are using our model in several varying application areas that are characterized by the need to view and navigate hierarchical information spaces, namely for browsing a file system, and for navigating and manipulating program structures and nested sets.

We give three examples of systems currently under development using this model: a 3D interpreter/browser for Bauhaus Linda, a multiset-based coordination language for human and process agents, characterized by hierarchical interaction spaces ([4]); a 3D interface for MAP, a visual programming environment ([3]); and a 3D interface for a hierarchical file system.

We believe that the structure of hierarchical information spaces lends itself well to a 3D interface environment in which the workspace gives the user a high-level view of the information, a sense of place within the information, and a sense of movement as the user examines and navigates the structure. There are many applications based on hierarchies that could benefit from 3D models of visualization and navigation. Our work in developing these 3D interaction environments is based on analysis of the needs of a user navigating an information space and attempting to work within this space, on the data itself and with other users.

The Model

Our development of this model was motivated by our need to interactively browse and manipulate recursive information structures from varying application areas, namely file spaces, multisets, and program structures. In each of our application areas, the information structures we wish to visualize and interact with are hierarchical and recursive: a file space consists of files and other file spaces, a Bauhaus multiset consists of primitives and nested Bauhaus multisets, and a MAP program consists of "map" structures, which are themselves constructed from basic values and "map" structures.

The recursive structure of each of our application domains led us to consider a graphically recursive representation of the information structures. Our model provides more than just information visualization; the user interacts with and manipulates data in a 3D recursive workspace. Users can move into structures, select objects, and perform application dependent functions on the information, all using a 3D editor.

Our model uses a cube object for containment, and to indicate *same level* definition and a relationship among objects. We chose cubes as the basic "container" building block, because cubes can be used to build well-defined geometrical arrangements and are well suited to nesting[2]. Other objects, such as 2D tiles with text or pictures, are used to indicate non-recursive, primitive units of information. Unfortunately, text is not feasibly manipulable by the user in 3D without sacrificing speed and usability, so text is entered into these 2D tiles using a standard 2D text widget. This is just one aspect of the speed/quality tradeoff that is still required for 3D interfaces.

The use of dimensions in 3D interaction environments varies. In our model, the horizontal axis or plane is used to indicate spatial arrangement or same-level definition; the vertical axis (depending on the specific application domain) is used to indicate either spatial or temporal arrangement; and the z axis is used to indicate nesting, containment, and level of depth. The issue of how to use spatial dimensions meaningfully in 3D environments is of particular concern in our research.

In some cases, multiple perspectives may be needed to provide the user with a sense of context in the information space. For example, giving a high-level "big-picture" view of the user's location can help prevent loss of context and help to maintain perspective. Systems like the File System Navigator[6] and Cone Trees[5] use such techniques effectively.

The technique of *zooming* is used in the model to balance detail and abstraction in the display. As the user zooms back from the information structure, a high-level view that gives a sense of place and context is shown, while zooming closer to the structure brings detail into focus and removes information about surrounding parts of the structure.

Of particular concern in interaction environments is how best to display information that is dynamic in nature. For example, if a user requests a directory listing of files in a shared directory, the user is only guaranteed that the listing reflects the state of the directory at the time the request is filled. This "snapshot" of the information can be user driven (the user makes the request to see the information at a given time), or can be updated dynamically in time intervals.

Applications

Bauhaus Linda

Bauhaus Linda is a coordination language which has been used as a foundation for building "Turingware" – collaborative systems in which humans and software agents are embedded in the same coordination framework. With the Bauhaus language, users and processes roam over, and interact with, a *coordination space* that is a hierarchy of multisets, each of which may contain primitive values (including processes and agents) and nested multisets.

We have applied our model of 3D interaction to this problem domain as a means of giving the Bauhaus user a visual browser/interpreter that allows navigation of, and dynamic interaction with, the Bauhaus coordination space. A browser window displays the elements of the "current" multiset: flat 2D tiles represent primitive elements and display either textual values or GIF images of human agents, and 3D cubes represent child multisets. The Bauhaus user can zoom backward to get a panoramic overview of the entire multiset contents, zoom up to tiles to see closeups of data or agents, pan a camera around a scene, and dive further in the Bauhaus hierarchy by zooming into selected multiset cubes. The 3D browser gives the Bauhaus user a sense of having a current virtual location in the Bauhaus hierarchy, and also a feeling of movement while navigating.

The Bauhaus Visual Browser has been built using the HOOPS 3-D graphics package and is currently running on a Sparc Classic workstation[1].

MAP

MAP is a visual programming environment in which the basic program constructs are represented by cubes in a 3D workspace. The workspace is used both as an editor for constructing programs, and as a visualization space for programs and their executions. Cubes are used to organize data within a program, to specify data structures and modules, and to impose a sequential execution of program statements. The user navigates within a program using up and down movements to change levels, and at a specific level, moves sideways to change nodes. As the user zooms closer to a particular region of the program and changes levels, less information about the regions on the previous level is shown, and more detailed information about the current region appears.

MAP programs are constructed using an interactive 3D editor built with *Open Inventor*^{TM}. Open Inventor is an object-oriented toolkit for developing interactive, three dimensional graphics applications. MAP is currently being implemented on a SGI Indigo and has also been tested on a SGI Onyx.

A Hierarchical File System

We have extended the desktop metaphor for file systems into a 3D interface by exploiting their nested nature, which can be directly represented in 3D space. As with the MAP environment, where nested structures in programs are represented with nested cubes, directory structures with several levels of sub-directories can also be represented with nested cubes.

Conclusion

We will present our hierarchical model and interface description and show examples of the systems in development. The issues of context, navigation and structure representation will be addressed.

References

1. *HOOPS Graphics System: Reference Manual*, 1994. Ithaca Software, Alameda, CA.
2. Glenn Franck and Colin Ware. Representing nodes and arcs in 3D networks. In *IEEE Symposium on Visual Languages*, pages 189–190, October 4-7 1994. St. Louis, Missouri.
3. Elisabeth Freeman and David Gelernter. In search of a simple visual vocabulary. Submitted to the *1995 IEEE Symposium on Visual Languages.*
4. Susanne Hupfer. Turingware. PhD Dissertation in preparation.
5. George G. Robertson, Stuart K. Card, and Jock D. Mackinlay. Information visualization using 3D interactive animation. In *Communications of the ACM*, volume 36, pages 57–71, April 1993.
6. Joel Tesler and Steve Strasnick. The file system navigator. Silicon Graphics.