The Challenge to Build Flexible User Interface Components for Non-Immersive 3D Environments

Raimund Dachselt
Dresden University of Technology
Multimedia Technology Group
D-01062 Dresden, Germany
dachselt@inf.tu-dresden.de

1 Introduction

Although 3D widgets, prototyping toolkits and interaction techniques were developed for Virtual Reality (VR) applications, more research is needed to overcome the conceptual and technological difficulties of 3D interface development. Immersive applications have not made their way to the mass market yet. However, one can notice enormous improvements in the graphics performance of PCs, pushing the development of non-immersive 3D graphics applications for the mass market, which use conventional input devices and displays. There is a huge potential for such applications especially in the field of marketing, sales and services. The use of 3D product data (from product development) for product presentations and retail stores suggests itself. Such systems for everyday users should not only be easy to use, but also have attractive interfaces to actually generate sales. These challenges are not easy to face with existing 3D interface technology, basically developed for immersive VR research systems. As a non-immersive example the next section shortly introduces a product presentation system. Based on experiences with the interdisciplinary development of this prototype the main section outlines problems with existing 3D widgets and toolkits and suggests principles for improved 3D interface design.

2 Example: 3D Product Presentation Environment

During postgraduate studies at the School of Art and Design Burg Giebichenstein Halle the author participated in an interdisciplinary research project called IMPLANTORIUM. This non-immersive system not only allows the presentation of

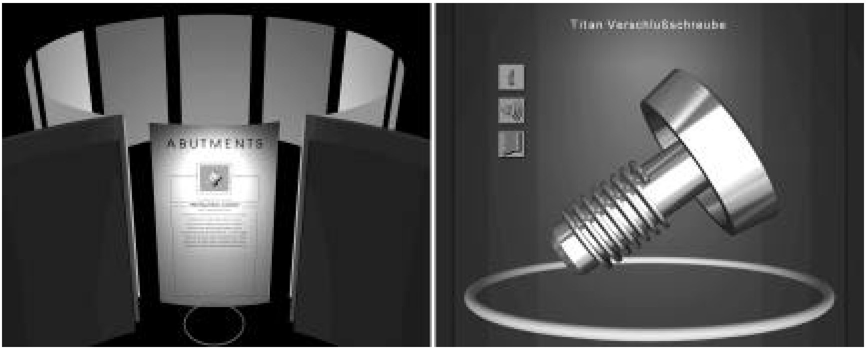

Figure 1: IMPLANTORIUM system: *Product Store* from above and *Product Showroom.*

the product range of an implant dentistry company, but also the interactive demonstration of product usage in surgery. It consists of two virtual revolving stages facing each other (Figure 1). The front stage can be rotated between the *product presentation* and *virtual surgery* state. The user always faces the back revolving stage, the *Product Store*, consisting of presentation columns. Each contains products of one particular category, like drills or abutments. After the desired product category was selected, the revolving stage is rotated to make the column facing the user. All products are listed on it with a preview picture and short description. The list can be scrolled and products can be selected, which appear in 3D, apparently being emitted by the preview picture. The finely rendered products can be rotated and zoomed using a Spacemouse or mouse. Additional product information may be displayed in this *Product Showroom*. In the *virtual surgery* state one can configure a surgery table, watch videos demonstrating surgery phases and interact with the products to simulate their use.

The system runs on SGI and Windows NT systems. C++ and the Open Inventor Toolkit (Strauss and Carey 1992) were used by means of intensively subclassing Inventor classes. Both interface elements and application objects (products) were implemented with the *nodekit* technology, thus adding semantics to subscene graphs and allowing flexible changes to object geometry and appearance. The nodekit definitions along with the Inventor data format made it possible to store actual object instances persistently in separate files. So it is possible to easily add new products or widget variants without further programming.

3 Suggestions for Improved 3D Interfaces

Three-dimensional widgets and toolkits were developed to create VR user interfaces and to support associated input devices. The dependence on special VR techniques prevents use in non-immersive applications and the standardization of 3D interface controls. A number of successful widgets were introduced to directly manipulate 3D data. Only few widgets, however, are available for applications without spatial relevance, where typical tasks to be solved are the selection of items from a list or the switching of states and – on a higher level – the representation of basic application metaphors and the integration of other widgets. In the following we suggest principles for better interface design.

Consider everyday products. Norman stresses the importance of visible clues provided by an everyday product, in particular affordances, constraints and mappings (Norman 1988). Unlike identically looking flat menu entries successful tools convey their functionality by means of their form and appearance, giving visual support on how to work with them. There is still the misconception among software developers, that esthetic quality is just a fancy addition to software. In fact it is a crucial factor for the usability of a system. Take 3D sliders as an example. Like many widgets they are often borrowed from 2D applications, possibly extruded and not adapted to the application space or underlying metaphor. They usually look like a collection of colorful building blocks, not effectively communicating their functionality. In contrast, Figure 2 shows a slider especially designed for the purpose of changing the transparency of an object. In the middle the slider looks opaque, gradually becoming more transparent to both edges, this way indicating its purpose. The form of the slider is adapted to the hemisphere in which models can be rotated.

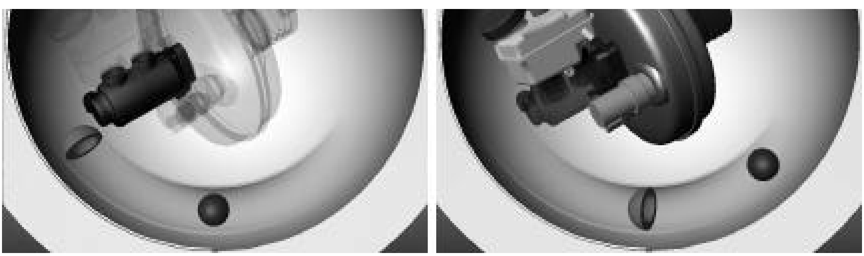

Figure 2: 3D-Slider to either isolate a part of an object or make it transparent

Work interdisciplinary. The slider example shows, that industrial designers are needed as experts in designing spatial relationships and balancing form and functionality. Although often discussed, interdisciplinary user interface development is by no means standard. It is obvious, that a computer specialist can't be at the same time a good designer or sound expert. Multidisciplinary teams, already common in multimedia production, are even more needed for VR due to the added dimension and complexity and the resulting level of design freedom. Combining the different disciplinary cultures of designers and engineers constitutes the key for success, and not ignorance on both sides. Toolkits supporting interdisciplinary interface development are needed. Take for example the “coordinate system” metaphor of the toolkit described in (Stevens et al. 1994). It is far too mathematical and technical for a designer, though working from a programmer’s point of view. Toolkits need a designer’s and programmer’s view, employing different metaphors and tools.

Provide action spaces. Tasks are usually associated with particular tools and certain spaces, e.g. the kitchen for cooking. This allows easy orientation and high efficiency, since tools are well arranged for a particular task. The Information Visualizer (Card et al. 1991) distributes information through a number of 3D and 2D rooms very much like real rooms through which the user can walk. We expand on the concept of information workspaces and introduce ***action spaces*** as 3D spaces with interface controls serving an associated task (the term was first coined by P. Kolbe, using the German word *Handlungsräume*). A number of tools and interface elements is laid out around a predefined viewpoint. Action spaces do not have to be rooms in a geometric sense. They are rather defined by the position of the user in the virtual environment and the interface objects inside the view frustum. The transition between action spaces is animated to facilitate orientation. The virtual *Product Store* constitutes one action space, the *Product Showroom* another. The first serves the product selection task, the latter the product examination. Selecting a product triggers the change of action spaces, visualized by two walls closing in front of the product column. They expose new interface controls associated with the information task. World-in-miniature widgets or other floating controls cause obscuration of objects and visual clutter (Mine 1996). This problem is solved by action spaces, where interface elements and application objects have their established constrained place. Orientation is simplified with this *task-centered navigation*, since the user only has to trigger change of places and not to move in some way.

Implement flexible user interface components. Research toolkits provide widget construction from very basic primitives assembled to form more complex widgets, although the least objects are mere collections of spheres, boxes etc. Moreover, they rarely allow flexible changes to a widget’s appearance or geometry, complicating the development of complex, non-primitive widgets. Our sample widgets are adaptable not only in terms of parameters like position, length or color, but also in their whole appearance or in parts of their geometry. They are implemented as independent Inventor *nodekits*, their complete behavior and constraints are predefined, so is the standard geometry and appearance. Every part, not only geometry data, but also textual descriptions, states and other fields, can be overwritten in definition files without programming. They are dynamically read in during system setup. So it is not only possible to exchange geometry parts, but also to remove them. Since parts are checked prior to interactions, one can also remove functional parts, thus for example degrading a product selection column to a mere display column. This top-down approach allows to start with metaphor-based widgets, which can be refined, adapted and also be assembled to form higher-level container components. The revolving stage is just one example, itself containing columns of various types, serving object selection and communicating application states. The widget flexibility allows to tailor interfaces to the corporate design needs of a company. *Conceptual constants* guarantee a degree of consistency, whereas perceptual, *audiovisual variables* create a distinctive interface, communicating a corporate or product identity and special mood. We call it *interface identity* and plead for a greater diversity of 3D user interfaces with high visual quality especially in the presentation and sales area. Standardization must not mean a stereotypic look and feel among all applications. Presentation columns with different shapes, colors or decorations will still be recognizable as presentation columns.

4 Conclusions and Acknowledgements

This paper has shown, that non-immersive 3D applications in the field of product presentations and sales demand user interfaces with flexible components. Suggestions were made to tackle the challenge of interface development in this area. We hope to stimulate discussion about 3D interface development with this approach, where flexible components are integrated in action spaces to make interaction tasks easier for casual users. Although the realized prototype was appreciated by both the company and test users, formal usability tests still have to be carried out. Especially the complex problem of appropriate high-level toolkits for interdisciplinary 3D interface development calls for further research.

I would like to thank the fellow students of the School of Art and Design Burg Giebichenstein Halle, especially Professor Peter Kolbe as the leader of our team. The interdisciplinary discourse and the joint development of the product presentation environment has stimulated some of the thoughts presented in this paper.

5 References

Card, S.K., Robertson, G.G., Mackinlay, J.D. (1991). The Information Visualizer, an Information Workspace. In *CHI '91 Conference Proceedings*, pp. 181-188, April 1991.

Mine, M.R. (1996). The Virtual World: Nice place to visit, but would you want to work there? In *Practical 3D User Interface Design* (Course Notes of the ACM SIGGRAPH 1996 conference), August1996.

Norman, D.A. (1988). *The psychology of everyday things.* New York, Basic Books. 1988.

Stevens, M.P., Zeleznik, R.C., Hughes, J.F. (1994). An Architecture for an Extensible 3D Interface Toolkit. In *UIST ’94 Proceedings*, pp. 59-67, Marina del Rey, CA, 1994.

Strauss, P.S., Carey, R. (1992). An object-oriented 3D graphics toolkit. *Computer Graphics* (SIGGRAPH ’92 Proceedings), 26(2), pp. 341-349, July 1992.