Three-Dimensional Widgets Revisited - Towards Future Standardization

Raimund Dachselt, Michael Hinz
Dresden University of Technology, Department of Computer Science
{dachselt, hinz}@inf.tu-dresden.de

Abstract

Three-dimensional user interfaces are not only used in specialized VR applications, but might play an important role in future desktop systems. Where only standard input devices are used, 3D widgets as geometric interaction elements are of special importance. This paper proposes a comprehensive classification of existing widget solutions by interaction purpose / intention of use. With the focus on widgets for system control including 3D menus it is especially suited for desktop 3D systems. In addition to that a consistent XML-based specification of 3D widgets facilitates application development and lays the ground for future standardization.

Keywords: *three-dimensional widgets, 3D user interfaces, widget classification, Desktop VR, Contigra*

1. Introduction and Motivation

Three-dimensional user interfaces (3D-UIs) are considered to be one of the future alternatives of present WIMP interfaces. More than a dozen years of research in the field of virtual reality (VR) have produced a rich variety of applications, novel input and output devices as well as 3D interaction techniques. VR applications are successfully used in well established domains basically by experts employing complex system settings. Though very specialized interaction techniques can be effective for solving problems by an expert, they hinder standardization across a variety of applications and devices. That is one reason why VR applications did not make their way into everyday life by now. Furthermore, the easy construction of 3D-UIs remains one of the big problems. Despite some commercial and research toolkits for building them, there are no development platforms comparable to sophisticated 2D user interface builders.

Desktop VR on the other hand does not use specialized hardware and employs common computer platforms. Although such 3D applications do neither allow full immersion nor very natural interactions, they seem to be promising for a broader variety of application domains including operating system interfaces. Certain tasks, such as pointing, selection, and symbolic input are easier to be accomplished in a desktop 3D environment. The recent introduction of high-performance PC graphics architectures, research 3D window managers and desktop projects such as Sun's Looking Glass [10], show the huge potential of 3D-UIs for the mass market.

How can interaction be done in desktop VR with standard input devices? One solution is the usage of three-dimensional widgets encapsulating geometry and behavior [2] as basic constituents of 3D applications. They play an important role especially for desktop VR, since they allow the subdivision of higher-dimensional interaction tasks into sub tasks being suitable for low-dimensional input devices such as mice. Thus 3D widgets can also be seen as interaction elements between traditional 2D and immersive 3D-UIs.

A huge number of diverse widgets have been developed over the past years mainly in VR research. In contrast to 3D interaction techniques and UI metaphors, where de facto standards are available, there exists neither a unifying classification nor a consistent description of 3D widgets until now. That is why widgets are often developed from scratch, even though comparable solutions already exist. This paper attempts to improve on that fostering the standardization of 3D widgets and thereby 3D-UI development.

2. Related Work

Most of the more than 200 existing 3D widget versions were developed within the context of an immersive 3D system or interaction technique. The lack of ready-to-use general purpose 3D widgets – as indicated already in [2] – still remains a problem. A few 3D widget libraries and toolkits exist, among them *it3d* [9]. However, its repertoire is mainly restricted to widgets borrowed from 2D.

In 1997 Leiner et al. presented a first overview on existing 3D widgets, roughly classifying them into geometric manipulation and application control [6]. A few other partial classification approaches exist, among them a subdivision of object manipulation widgets into linear, non-linear and high-level manipulation as introduced by Watt and Policarpo in [11].

3D interaction techniques can be seen as widgets without always having a geometric representation. Various researchers, among them Bowman, LaViola, Mine, Poupyrev and others have developed a huge variety of techniques. Many of them are documented in [1]. Techniques for selection, manipulation, travel, wayfinding, and system control are classified mostly in terms of task decomposition. Figueroa et al. presented a more practical XML-based specification and classification of interaction techniques with *InTML* [5]. However, these classifications are focused on 3D interaction techniques and not explicitly on 3D widgets with associated geometry. All mentioned classifications were considered and partly integrated into this work.

3. A Widget Classification Scheme

The widget classification scheme introduced here is taken from the doctoral thesis by one of the authors [3]. The focus lies on classifying interactive 3D widgets, which are general enough to be used in various 3D projects and meet the criteria of a geometric representation as postulated by Conner et al. [2].

Finding appropriate and unifying classification criteria is a difficult task, since certain widget groups have certain peculiarities. Moreover, mathematical and purely theoretical classifications would be less of real help for the 3D-UI developer. A number of criteria was considered, among them *application area* (too general), *interaction task* (sometimes too technical and overlapping), *input device / degree of freedom* (useful for e.g. tablet-associated widgets, but too restrictive), and *optical appearance* (interesting in terms of space requirements or geometric shape). Finally the critera *interaction purpose / intention of use* was chosen for a unifying basic classification scheme. The sub classification may vary depending on more suitable criteria such as type of the manipulated data. Table 1 depicts the four main categories and all subdivisions. All listed entries are only categories, where particular 3D widgets of each group are omitted due to space limitations. They can be viewed online [8] using an interactive 3D widget hierarchy containing more than 70 classified widget types (see also [3] for details).

Table 1. 3D widget classification categories

Direct 3D Object Interaction		
	Object Selection	
		Direct Selection
		Occlusion Selection
		Distance Selection
	Geometric Manipulation	
		Linear Transformation
		Non-linear Transformation
		High-Level Object Manipulation
3D Scene Manipulation		
	Orientation and Navigation	
		Direct Viewpoint Selection
		Guided Transport
		Using Miniatures
	Scene Presentation Control	
		Light Manipulation
		Camera Manipulation
		Sound Control
Exploration and Visualization		
	Geometric Exploration	
	Hierarchy Visualization	
	3D Graph Visualization	
	2D-Data and Document Visualization	
	Scientific Visualization	
System / Application Control		
	State Control / Discrete Valuators	
		Activation
		Two States
		Multiple States
	Continuous Valuators	
		Scalar Values
		Multiple Values
	Special Value Input	
		Color Chooser
	Menu Selection	
		Temporary Option Menus
		Single Menus
		Menu Hierarchies
	Containers	

Direct 3D Object Interaction is a category containing well-known widgets and interaction techniques. Examples are 3D-Cursor, Silk Cursor, and Image Plane Selection falling into the subgroups of *Object Selection* as well as Interactive Shadows, Rack Widget, Repelling Tool etc. into those of *Geometric Manipulation*.

3D Scene Manipulation contains two categories. *Orientation and Navigation* and its subgroups integrate widgets such as Flying Chair, Worlds in Miniature, and Magic Mirror. *Scene Presentation Control* unites existing widget solutions for light, camera, and sound control.

Exploration and Visualization constitutes a very extensive group, since various 3D techniques exist in this field. However, many of them are very domain-specific, do not provide interactions and are therefore excluded. Five subgroups classify more than 20 widgets. *Geometric Exploration* contains generic techniques, such as Cutting Plane, 3d Magic Lenses etc., *Hierarchy Visualization* Information Cubes, Cone Trees, Cam Trees, and others, *3D Graph Visualization* various 3D solutions, e.g. NestedVision3D, *2D-Data and Document Visualization* Perspective Wall, Document Lens etc., and finally the subgroup *Scientific Visualization* Rake, Hedgehog, and similar widgets.

System / Application Control is a category representing important widgets for almost every 3D application. Whereas the subgroups *State Control / Discrete Valuators, Continuous Valuators,* and *Special Value Input* contain 3D widgets partly well-known from 2D interfaces, the subgroups *Menu Selection* and *Containers* list very 3D specific solutions having rarely a counterpart in 2D. Containers comprise spatial solutions for integrating other widgets, e.g. Toolspaces or Hand-Held Windows.

Table 2. Classified 3D menu widgets

System or Application Control			
	Menu Selection		
		Temporary Option Menus	
			Rotary Tool Chooser
			Menu Ball
			Command & Control Cube
			Popup Menu
			Tool Finger
			TULIP
		Single Menus	
			Ring Menu
			Floating Menu
			Drop-Down-Menu
			Revolving Stage
			Chooser Widget
			3D-Palette, Primitive Box etc.
		Menu Hierarchies	
			Hands-off Menu
			Hierarchical Pop-Up Menus
			Tool Rack
			3D Pie Menu
			→ Hierarchy Visualization

To provide an example of specific 3D widgets arranged in subgroups of the classification, table 2 depicts the classification of widgets for *Menu Selection*. System or application control widgets are of special importance for most 3D applications and especially desktop VR. As Bowman et al. state in [1], system control for 3D-UIs is only in its infancy as a research topic. The classification of 3D menu techniques into the subgroups *Temporary Option Menus, Single Menus,* and *Menu Hierarchies* facilitates the choice of appropriate solutions. As with all classified widgets, similar solutions were reconciled and presented as one singular widget type.

4. Consistent Specification of 3D Widgets

The classification approach presented here allows 3D application programmers to choose parts of the user interface in terms of functionality and intention of use. Requests like "I need widgets to non-linearly transform objects, to input continuous scalar values, and a simple 3D option menu" can be answered with appropriate widget solutions. And yet, the classification is only one part. Take for example a 3D developer having identified a ring menu [7] as the appropriate choice for an application. As depicted in figure 1, there are still many different solutions for ring menus in terms of visual appearance, functionality, and implementation.

To facilitate standardization while maintaining optical flexibility and allowing well-defined configuration options, 3D widgets must be specified in a consistent manner. Within the project CONTIGRA a declarative component architecture was developed along with a special XML-Schema language for defining interfaces of 3D widgets / components [4]. It allows the definition of common metadata for each widget type including name, description, author, publication, and picture. The main part defines a set of high-level parameters, which describe the functionality and configuration options for each widget in a declarative way.

Table 3. Parameters of the ring menu widget

Parameter Semantics	**Parameter**
General	ItemList
	SelectedItem
Geometry	RotateLeftGeometry
	RotateRightGeometry
	FixedGeometry
	InterItemGeometry
	SelectionGeometry
Appearance	ItemRatio
	RingRadius
Behavior	RotationSpeed
Audio	-

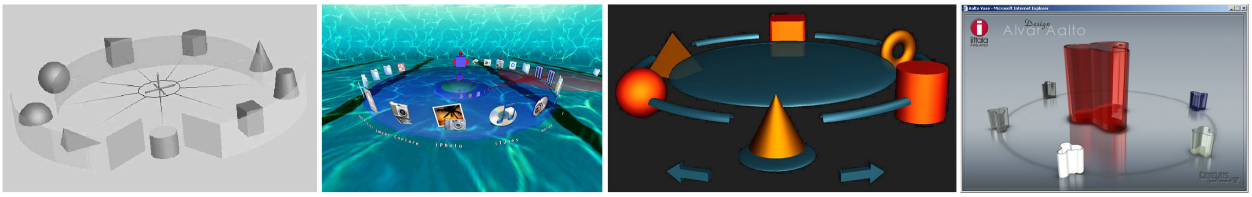

Figure 1. Various ring menus: from the original paper [7] (left) and other projects [3]

Table 3 lists the parameters of the ring menu example. Parameters possess several attributes, among them name, data type, description, event/configuration information, and their semantics as shown in the first column. This way 3D widgets can be specified in a consistent declarative way, as can be seen in the data sheets for each widget in the online classification [8]. Note, that basically system control widgets were specified by now, as they are the focus of this work.

The specification of each 3D widget, i.e. an XML file, is very flexible, since it can be extended and transformed to any kind of description or data sheet. Even more important, it can be transformed (e.g. using XSLT) to any kind of implementation file, say a C++ header or Java interface. This way, widgets can be implemented depending on a specific 3D toolkit but using a comprehensive and standardized specification across applications. This was successfully demonstrated with the CONTIGRA architecture as one possible development platform in the field of desktop 3D applications [3],[4]. Many of the widgets were already implemented, used in complex sample applications, and can also be tested online.

5. Conclusion and Future Work

This paper presented an extensive classification approach for interactive three-dimensional widgets possessing a geometric appearance. The chosen criteria *interaction purpose / intention of use* shall facilitate the practical use of widgets in 3D projects. An XML-based specification language allows the consistent specification of widgets in terms of configurable parameters. Especially with the advent of desktop 3D applications for the mass market standardization efforts become important, which are comparable to the field of 2D user interfaces.

This work is intended to stimulate discussion among 3D-UI researchers. The proposed classification must be refined and extended, alternative classification schemes are imaginable and potentially desirable. The existing XML specification language could be extended as well and possibly merged with description schemes for 3D interaction techniques such as InTML [5]. Our vision is a repertoire of well-defined, standardized 3D widgets and 3D interaction techniques to be used across applications in the field of immersive VE, augmented and mixed reality, and desktop 3D.

6. References

[1] Bowman, D.A., Kruijff, E., LaViola, J., Poupyrev, I., *3D User Interfaces: Theory and Practice*, Addison Wesley, Boston, July 2004.

[2] Conner, D.B., Snibbe, S., Herndon, K.; Robbins, D., Zeleznik, R., van Dam, A., "Three-Dimensional Widgets", *Proceedings of the ACM Symposium on Interactive 3D Graphics*, ACM Press, New York, March 1992, pp. 183-188.

[3] Dachselt, R., *Eine deklarative Komponentenarchitektur und Interaktionsbausteine für dreidimensionale multimediale Anwendungen* (doctoral thesis, in English: *A Declarative Component Architecture and Widgets for 3D Multimedia Applications*), Der Andere Verlag, Tönning, December 2004.

[4] Dachselt, R., M. Hinz, and K. Meißner, "CONTIGRA: An XML-Based Architecture for Component-Oriented 3D Applications", *Proceedings of the ACM Conference on 3D Web Technology (Web3D 2002)*, ACM Press, New York, February 2002, pp. 155-163.

[5] Figueroa, P., Green, M., Hoover, H.J., "InTml: A Description Language for VR Applications", *Proceedings of the ACM Conference on 3D Web Technology (Web3D 2002)*, ACM Press, New York, February 2002, pp. 53-58.

[6] Leiner, U., Preim, B., Ressel, S., "Development of 3D-Widgets – An Overview", *Proceedings of Simulation and Animation SCS Europe*, Erlangen, March 1997, pp. 170-188.

[7] Liang, J., Green, M., "JDCAD: A Highly Interactive 3D Modeling System", *Computers and Graphics*, Vol. 18/4, 1994, pp. 499-506.

[8] Online Widget Classification: www.3d-components.org

[9] Osawa, N., Asai, K., Saito, F., "An Interactive Toolkit Library for 3D Applications: it3d", *Proceedings of the 8th Eurographics Workshop on Virtual Environments,* Eurographics Association, May 2002, pp. 149-157.

[10] Sun Microsystems – Project Looking Glass: http://www.sun.com/software/looking_glass/

[11] Watt, A., Policarpo, F., *The Computer Image*, ACM Press, New York & Addison-Wesley, 1998.