

3D Graphical User Interfaces

Farid BenHajji

Erik Dybner

Department of Computer and Systems Sciences

Stockholm University and

The Royal Institute of Technology

July 1999

ABSTRACT

Human-computer interaction currently faces the challenge of taking advantage of new technologies, which efficiently use the capabilities of the computing systems of today and more effectively match human capabilities and perception.

We investigate how 3D Graphical User Interfaces (GUI) can be useful in software applications of today and identify general areas where 3D technologies can be more useful than other technologies. This paper consists of a survey of 3D standards, GUIs and in particular 3D GUIs. A 3D GUI prototype based on applications of ABB Network Partner AB is described.

There are several possible advantages using 3D graphics. Interesting aspects are the possibilities of natural, intuitive navigation and interaction with computer systems. However, there are a couple of problems. The lack of commonly accepted standards for 3D interfaces e.g. widgets and graphic formats is one of them.¹

Keywords

Human-Computer Interaction (HCI), Three-dimensional (3D), Graphical User Interface (GUI)

¹ This thesis corresponds to 20 full-time working weeks per author.

1	INTRODUCTION.....	1
1.1	BACKGROUND.....	1
1.2	EMPLOYER.....	1
1.3	PURPOSE.....	1
1.4	METHODOLOGY.....	1
1.5	RESPONSIBILITIES.....	2
1.6	STRUCTURE OF THE THESIS.....	2
1.7	ACKNOWLEDGEMENTS.....	2
2	COGNITIVE ASPECTS ON USER INTERFACE DESIGN.....	3
2.1	NAVIGATION.....	3
2.1.1	<i>Cognitive Maps</i>	3
2.1.2	<i>Spatial Ability</i>	4
2.2	INTERACTION.....	4
2.2.1	<i>Attention</i>	4
2.2.2	<i>Affordances</i>	4
2.2.3	<i>Abstraction</i>	5
3	INTRODUCTION TO USER INTERFACES.....	6
3.1	CHARACTER-BASED INTERFACES.....	6
3.2	2D GRAPHICAL USER INTERFACES.....	6
3.2.1	<i>Desktop Metaphor</i>	7
3.2.2	<i>Interface Components</i>	7
4	3D GRAPHICAL USER INTERFACES.....	8
4.1	PERCEPTION OF DEPTH.....	8
4.1.1	<i>Monocular Depth Cues</i>	8
4.1.2	<i>Binocular Depth Cues</i>	9
4.2	INTERFACE COMPONENTS.....	10
4.3	INPUT DEVICES.....	12
4.4	OUTPUT DEVICES.....	13
4.5	VIRTUAL REALITY.....	14
5	APPLICATION AREAS FOR 3D GRAPHICAL USER INTERFACES.....	16
5.1	INFORMATION VISUALISATION.....	16
5.2	SIMULATION.....	17
5.2.1	<i>Virtual Wind Tunnel</i>	17
5.2.2	<i>Flight Simulator</i>	17
5.3	COMPUTER AIDED DESIGN.....	18
5.4	COMPUTER SUPPORTED COOPERATIVE WORK.....	19
5.5	ENTERTAINMENT.....	20
5.6	MARKETING.....	20
6	OVERVIEW OF 3D GRAPHICS STANDARDS.....	21
6.1	OPENGL.....	21
6.1.1	<i>Introduction</i>	21
6.1.2	<i>History</i>	21
6.1.3	<i>Design</i>	22
6.1.4	<i>Architecture</i>	23
6.1.5	<i>Application Programming Interface Specification</i>	23
6.1.6	<i>Libraries on top of OpenGL</i>	23
6.1.7	<i>Applications</i>	24
6.2	VIRTUAL REALITY MODELLING LANGUAGE.....	24
6.2.1	<i>Introduction</i>	24
6.2.2	<i>History</i>	25
6.2.3	<i>Design</i>	25
6.2.4	<i>Presentation and Interaction</i>	26
6.2.5	<i>External Authoring Interface</i>	26
6.2.6	<i>Problems</i>	27

6.2.7 *Future*..... 28

6.3 CONCLUSION 28

7 COMPARISON OF 2D AND 3D GRAPHICAL USER INTERFACES 29

7.1 NAVIGATION..... 29

7.1.1 *Cognitive Maps* 29

7.1.2 *Spatial Ability* 29

7.1.3 *Motion Sickness* 29

7.2 INTERACTION 30

7.2.1 *Attention*..... 30

7.2.2 *Affordance*..... 30

7.2.3 *Abstraction*..... 30

7.2.4 *Input and Output*..... 31

7.3 DISCUSSION 31

8 ANALYSIS..... 32

8.1 CURRENT SITUATION 32

8.2 DESIRED SITUATION 32

8.2.1 *Information Visualisation* 33

8.2.2 *Attractiveness*..... 33

8.3 NEW USER INTERFACE DESIGN 33

8.3.1 *Navigation*..... 33

8.3.2 *Interaction*..... 35

8.4 SYSTEM ARCHITECTURE 36

9 SUMMARY 37

BIBLIOGRAPHY 38

APPENDIX A 41

APPENDIX B 46

1 Introduction

This chapter gives the reader the background and purpose of the thesis and describes our methodology

1.1 Background

The rapid development of computer hard- and software has caused significant changes in the way users interact with computers. We have seen a development from the early stages with character based interfaces to modern Graphical User Interfaces (GUI) of today.

The Windows, Icons, Menus, and Pointer (WIMP) interface paradigm dominates the computing systems of today, these interaction techniques were originally developed for machines that are now 10 or nearly 20 years old. Human-computer interaction (HCI) currently faces the challenge of evolving past the WIMP plateau and taking advantage of new technologies, which efficiently use the capabilities of the computing systems of today and more effectively match human capabilities and perception.

Until recently, Three Dimensional (3D) graphics were primarily used in specialised domains such as Computer Aided Design (CAD) and flight simulators. Due to the increasing performance of new computer hardware, in combination with state-of-the-art interactive 3D computer games, e.g., the category of “Doom”-like games [Lampton 1994], there has been a rapid growth in interest for real-time 3D graphics [Mullet et al 1995].

1.2 Employer

Asea Brown Boveri (ABB) is a Swedish/Swiss industrial enterprise and a market leader in producing power stations, power transmissions, and power transport. It is also active in other areas of industry. ABB Network Partner AB is a subsidiary that among other things produce information management systems and protection and control terminals for power stations.

ABB Network Partner AB is like many other companies, very interested in how 3D graphics technologies can be applied to improve their processes and products.

1.3 Purpose

The purpose of the thesis is twofold: to conduct a survey of 3D GUIs from a cognitive perspective and to investigate how to implement 3D GUIs.

We will therefore investigate what the term 3D GUI implies and identify current application areas. Further, we will discuss how 3D GUIs can be applied to improve company processes and products.

To accomplish our second goal we will study 3D graphics standards in order to investigate what it takes to implement 3D GUIs. Finally, we intend to implement a prototype to show how ABB Network Partner can take advantage of 3D GUIs in their applications, which will be our main contribution to the field.

1.4 Methodology

Our work consists of a survey of GUIs and in particular 3D GUIs. There are several papers in this area of research on which we have based our results. Many of these papers are only available on the Internet due to this being a newly developed research area. The parts of the

thesis covering 3D standards are based on studies of the reference literature of the standards and our own experience of using them. Finally we have studied the user interface of different ABB Network Partner AB applications, for which we have constructed a 3D GUI prototype, to show how their products can be improved by 3D.

1.5 Responsibilities

The contents of chapter 2, chapter 5, chapter 7 and chapter 8 are under the responsibility of Farid BenHajji, and the contents of chapter 3, chapter 4, chapter 6, and chapter 9 are under the responsibility of Erik Dybner. We have thoroughly discussed and jointly developed all parts of the thesis.

1.6 Structure of the Thesis

We start off in chapter 2 by describing the cognitive aspects on user interface design. In chapter 3 we will give a brief introduction to user interfaces and in chapter 4 an introduction to 3D graphical user interfaces. Chapter 5 is a survey of application areas for 3D Graphical User Interfaces followed by chapter 6 which gives an overview of 3D graphics standards. Chapter 7 includes a comparison of 2D and 3D graphical user interfaces based on the cognitive aspects introduced in chapter 2. Chapter 8 describes the current and wanted situation for the user interfaces of the applications of ABB Network Partner which leads to a description of a new user interface. Finally chapter 9 sums up the thesis.

1.7 Acknowledgements

This thesis originated from a commission given by ABB Network Partner AB.

We would like to thank our supervisor at DSV and ABB Network Partner AB, Johan Kummeneje. We would also like to thank Jan Bystedt who was kind enough to help proof read this thesis.

2 Cognitive Aspects on User Interface Design

User interface design is an interdisciplinary activity in which research from the field of psychology can be of great use. An understanding of how humans perceive and interact with their environment can contribute to the design of more intuitive user interfaces. Therefore, this chapter is a short introduction to some cognitive aspects of navigation and interaction.

2.1 Navigation

Navigation is the process by which people control their movements using environmental cues and artificial aids such as maps so that they can reach their goals without getting lost. A major problem for users in all forms of navigation is how to maintain knowledge of their location and orientation while moving through space.

Important aspects when discussing navigation in computer systems are cognitive maps and spatial abilities. [Darken and Sibert 1993] have shown that people use physical world wayfinding strategies in large virtual worlds, i.e. cognitive maps for navigational aid. According to [Höök and Dahlbäck 1997] individual differences in spatial abilities affect users' ability to navigate in computer systems.

2.1.1 Cognitive Maps

Birds has often proven to be superior navigators during migration and although research has not produced definitive answers, there exists evidence that indicates the use of advanced perceptual abilities to detect and encode environmental cues [Darken and Sibert 1993].

It is believed that birds use a landmarking technique in which a cognitive map is created [Darken and Sibert 1993]. Landmarks on this map may be visual, acoustic, or olfactory. Resources such as shelter and water is encoded along with information concerning the shortest and safest paths between points. Increased altitude enables spatial relationships to be refined since more landmarks can be seen simultaneously, but it also decreases the strength of the stimuli.

In similarity with birds humans are also thought to form cognitive maps of their environments for use in navigation, using a set of generic components such as paths, edges, landmarks, nodes, and districts. According to [Darken and Sibert 1993]they include:

- Paths: linear separators, walkways and passages.
- Edges: linear separators, walls or fences.
- Landmarks: objects that are in sharp contrast to their immediate surroundings, such as a church spire.
- Nodes: sections of the environment with similar characteristics. For example, a group of streets with the same type of light posts.
- Districts: logically and physically distinct sections.

According to [Chen and Czerwinski 1998] simply adding cues like borders, boundaries, and gridlines in a user interface significantly improve navigation performance.

2.1.2 Spatial Ability

It has been shown that users' spatial ability is relevant to their ability to make use of computer systems. There exists enough knowledge of individual differences and navigation to say that these are important and significant [Höök and Dahlbäck 1997].

When discussing spatial ability three various dimensions are commonly addressed [Satalich 1995]. They are spatial orientation, spatial visualisation, and spatial relations. Spatial orientation involves the ability to mentally move or transform stimuli, while retaining their relationships. It also involves the mental manipulation of an object using oneself as a reference. Spatial visualisation goes further, in that the person can manipulate the relationships within an object. Finally, spatial relations, consists of the ability to imagine how an object will align from different perspectives.

In [Höök and Dahlbäck 1997] it is concluded that spatial ability affects user ability to navigate large file structures. Similar results are also found concerning navigation in hypermedia, i.e. a collection of non-linear text-based nodes that are linked together. The best user was almost 20 times faster than the slowest in using such a system to solve six problems.

2.2 Interaction

Interaction deals with how humans exchange information with the environment, typically why we choose to focus on certain elements (attention), how we know what we can do with these elements (affordances) and how we filter redundant information (abstraction).

2.2.1 Attention

The attention of humans is among other things attracted to differences in visual appearance, e.g. colour, shape, texture, shading, size, location, and the ability to register movement [Wiss and Carr 1998].

The location of an object also determines the amount of attention we give to it. Objects located in the centre of the visualisation tend to get more attention than peripheral objects. Also, if our attention is focused on something particular, elements close to this will get more attention than elements further away. Another thing is our ability to register movement, which among other things makes blinking objects catch our eyes, making it difficult to concentrate on other things. This ability is inherited from our ancestors for whom it was very important to notice movement that could be either danger or potential food in their environment [Smith 1993].

2.2.2 Affordances

Affordance refers to the perceived and actual properties of the things, primarily those fundamental properties that determine just how things can possibly be used [Norman 1988]. In other words, affordance is the cues that an object gives to explain what can be done with it. A button affords pushing, a knob affords turning, and a ball affords bouncing or throwing. What the object affords to a person relates to that person's current mental model, cultural aspects, etc.

A psychology of causality is also at work as we interact with the environment, meaning that the perceived result of an action appears to be caused by a particular action. The failure and the action may very well be related only through a coincident. This kind of false causality may occur in user interfaces due to poor design [Norman 1988].

2.2.3 Abstraction

The amount of data that flow through the human senses is so large that we are forced to filter much of it, in order to focus on elements that we consider being important. The problem is that filtering removes global context and may leave us disoriented. One solution to retain global context is to organise the data that we do retain in ways that promote understanding at the expense of detail. Abstraction implies a many-to-one transformation between the possible variants of a feature to a single invariant form. This can be illustrated by the gestalt law of perception [Smith 1993].

The gestalt law of perception is one way to describe how the human brain relates visual symbols, by grouping symbols that are close together and forms structure that is only suggested in the visualisation. Gestalt psychology is a school of psychology that emphasises patterns of organisation. “gestalt” is a German word, which translates to “form, shape, pattern, or organised whole”. The basic belief is that the whole is greater than the sum of its parts. The gestalt psychologists catalogued a number of laws of perceptual grouping which provide evidence for the nature of low-level visual processing, as well as a clue to the nature of the visual representation. They suggest that people group and interpret stimuli in accordance with four laws of perceptual organisation: proximity, similarity, closure and continuity [Smith 1993].

The principle of proximity suggests that elements closer together, tend to be grouped together as opposed to those further away. Objects that are grouped together based on similar properties apply to the principle of similarity. Another law of perceptual organisation is called closure, it includes the ability to see apparently disorganised or unrelated parts as a meaningful whole and to construct a total picture from incomplete or ambiguous material.

The last gestalt principle of organisation, continuity deals with objects that define smooth lines or curves. Good continuation is such a powerful organising principle that when the figure is restored to its initial state, you may “lose” the new organisation you just found as your original organisation takes over again. Similarity, Proximity, and closure are described in figure 2-1.

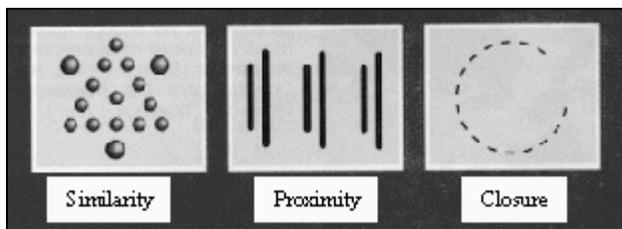


Figure 2-1 Three laws of perceptual organisation: proximity, similarity, and closure [Smith 1993].

3 Introduction to User Interfaces

The user interface is one of the most important parts of any program because it is the only connection between the system and the user. Up until now there has been basically two different ways to communicate with the interfaces; either by typing commands in a character-based interfaces (CBI) or by pointing and clicking on objects in a GUI. CBIs are generally more difficult to learn but is often considered to be more flexible than GUIs [Preece et al. 1994].

We will give a brief introduction to user interfaces and present different types of user interfaces, starting with the CBI and the two dimensional (2D) GUIs. Next chapter introduces 3D GUIs.

3.1 Character-Based Interfaces

The display screen of a CBI is typically represented as an array of boxes, each of which can hold only one character. Since a CBI can only display characters, the users must interact with the system by typing commands on a keyboard (See figure 3-1).

The basic input device on most small computer systems is a keyboard. As characters are typed, they are stored in the memory and copied to a basic output device, a display screen.

CBIs can be concise and unambiguous, but are often difficult for a novice to learn and remember mainly because it is essential for the users to learn the syntax. Once it is learned a CBI can often be flexible and powerful for advanced users. However, error rates are typically high, training is necessary, retention may be poor and error messages can be difficult to understand [Preece et al. 1994].

Many character-based interfaces include some features of GUIs, such as menus. Other character based interfaces include shapes for drawing simple objects but are not capable of representing more complicated objects e.g. curves; these are sometimes called graphical character based user interfaces to separate them from true GUIs [Preece et al. 1994].

```
Microsoft<R> MS-DOS<R> Version 6.2
<C> Copyright Microsoft Corp 1981-1993.

C:\>copy g:\supervisor\final_thesis.doc a:\cheater\
C:\>
```

Figure 3-1 A Character Based Interface.

3.2 2D Graphical User Interfaces

Fundamental for the GUIs is that the screen is divided into a grid of picture elements (pixels), and by turning the pixels on or off pictures can be formed [Davis 1991].

Similar to Character Based Interfaces, GUIs also uses a keyboard as an input device and a display as an output device. With the introduction of the desktop metaphor (see section 3.2.1), a new input device that revolutionised the interaction was introduced. The mouse is a small

device that moves the cursor relative to its present position and makes it easy to interact with the computer system [Smith et al. 1982].

GUIs usually features windows, icons, menus, and pointers. The window makes it possible to divide the screen into different areas and to use each area for different tasks. Each window can contain icons, which are small pictures of system objects representing commands, files or windows. Menus are a collection of commands and objects from where a user can select and execute a choice. The pointer is a symbol that appears on the display screen, that you move to select and execute objects and commands.

A GUI often includes well-defined standard formats for representing text and graphics, thus making it possible for different programs using a common GUI to share data.

The use of GUIs helps reduce the mental effort required to interact with the programs. Instead of remembering sequences of complex command languages, users only need to learn how to interact with the simulated world of objects, e.g. icons and menus [Preece et al. 1994].

3.2.1 Desktop Metaphor

One of the most famous of the first graphical user interfaces is called the Star user interface and was designed by Xerox Corporation's Palo Alto Research Centre in the 1970s. Yet it was not until the early 1980s that the development of the Apple Lisa and the Macintosh made graphical user interfaces popular [Smith et al. 1982].

The Star interface is based on the physical office and thus designed to be more like the physical world that already is familiar to the users [Smith et al. 1982]. The idea of the interface metaphor was to create electronic counterparts to the physical objects in an office. This involved representing office objects, e.g. papers, folders, filing cabinets, and trashcans, on the screen. The overall organising metaphor presented on the screen was that of a desktop, resembling the top of an office desk.

Some of the principles that were developed with the Star computer included direct manipulation, What You See Is What You Get (WYSIWYG), and consistency of commands [Smith et al. 1982]. Direct manipulation is a communication style where objects are represented on the screen and can be manipulated by the user in ways analogues to how the user would work with the real objects. WYSIWYG is a way to represent the user's conceptual model in a form that can be displayed. For example a document is displayed on the screen, as it would look in printed form [Preece et al. 1994].

3.2.2 Interface Components

Interface components (sometimes known as widgets) are graphical objects with behaviour that are used to control the application and its objects [Conner et al. 1992]. The objects, such as buttons, menus, sliders, texts, labels, droplists, and lists are combined to make up the GUI. They all allow user interaction via a pointer device and/or a keyboard.

4 3D Graphical User Interfaces

In the literature the term “3D interface” is used to describe a wide variety of interfaces for displaying and interacting with 3D objects. True 3D interfaces i.e. interfaces with all its components in a 3D environment have not yet had any major impact outside the laboratory. The interfaces nowadays referred to as 3D GUIs are almost exclusively “hybrids” between 2D and 3D interfaces. In this thesis we will refer to these "hybrid" interfaces simply as 3D GUIs.

In the early days of 3D computing the only users of 3D graphics were exclusive groups of scientists and engineers (i.e. experts) with access to expensive super computers and workstations. Today, the growth in platform-level support for real-time 3D graphics has increased the interest in adding to the dimensionality of the GUIs [Mullet et al 1995]. The user group has grown from an exclusive group of expert users, to a diverse group of researchers and practitioners in such areas as CAD/CAM, medicine, engineering, etc. This imposes high requirements on the user interface since it is now going to be used by a broad heterogeneous user group with a big diversity in knowledge of computing. One possible way to accomplish this is to design the user interface in a way that makes it possible for users to engage more natural skills in navigating and orientating themselves within applications. The concept Virtual Reality (VR), also known as Virtual Environment (VE), is an attempt to accomplish this. It is also a field closely coupled to 3D GUIs.

4.1 Perception of depth

An important topic in the field of 3D graphics is how to visualise objects so that the users get a good impression of their shapes as well as their relative positions. To accomplish this it is important to consider how depth and shape are perceived by human beings.

How does one tell from a two dimensional image where objects are located in space? In a normal situation this is nothing we have to worry about since our brain takes care of all that for us by using the information it obtains from the retina. However to be able to create credible 3D graphics it is necessary to have a deeper insight into these matters.

The retina receives information in two dimensions but the brain translates these cues into three-dimensional perceptions. It does this by using both *monocular cues* (which require only one eye) and *binocular cues* (which require both eyes) [Smith 1993].

4.1.1 Monocular Depth Cues

The monocular depth cues are relevant when discussing computer graphics since these are the cues that will have to be used to create the illusion of depth on a normal two-dimensional display. These depth effects are experienced just as powerful with one eye closed as when both eyes are used.

With clever use of monocular cues it is possible to create images whose individual parts make sense but whose overall organisation is impossible in terms of our existing perceptual schemas. The artist Oscar Reutersvärd skilfully used this in the creation of his famous “impossible figures” (see figure 4-1) [Gabriel 1985].



Figure 4-1 Swedish artist Oscar Reutersvärd cleverly manipulated monocular depth cues to create his “impossible figures” [Gabriel 1985].

Imagine standing on a straight railroad track watching the rails recede in the distance. The two rails will seem to get closer together the further away you look. This illustrates an important monocular cue known as linear perspective.

Other more obvious monocular cues include decreasing size, i.e. objects further away from us produce a smaller image than objects close to us and the fact that objects further away appear in a higher horizontal plane (i.e. objects in the distance appear closer to the horizon). Texture and clarity also affect the way we perceive depth, we can see objects close to us more clearly and detailed than objects far away, also the texture of the objects is more fine grained in the distance.

Finally just the effective use of patterns of light and shadow can provide important depth information. All the monocular depth cues are summarised in figure 4-2.

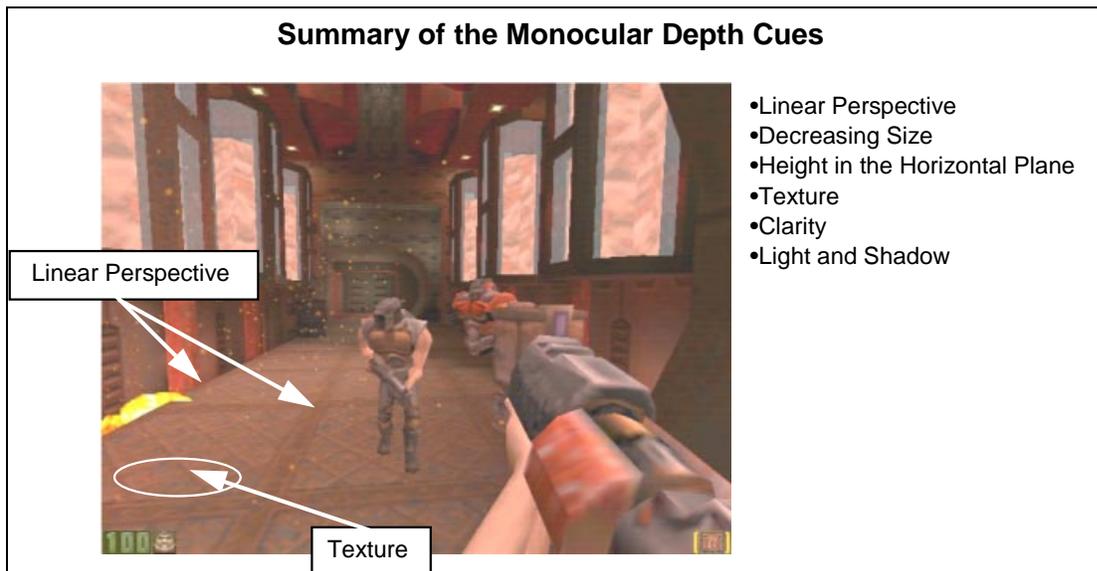


Figure 4-2 The computer game Quake II uses monocular depth cues to provide a realistic game world.

4.1.2 Binocular Depth Cues

Earlier discussed monocular depth cues are not as convincing as the much richer depth effect that occurs when both eyes are used. The appearance of depth that results from the use of both

eyes is called stereopsis [Smith 1993]; it arises when the brain combines the two slightly different images it receives from each eye into a single image with depth (See figure 4-3). The effect of stereopsis can be achieved artificially by using devices like Head Mounted Displays or shutter glasses, as described in section 4.4. As opposed to monocular depth cues 10 % of people can not interpret stereopsis, i.e. they can not detect binocular depth cues, thus making them ‘stereoblind’ [Media Lab 1998].

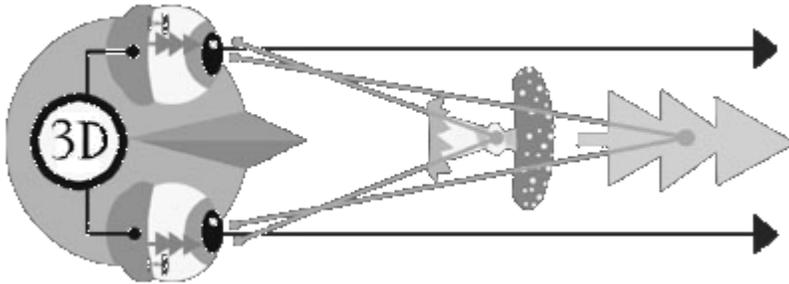


Figure 4-3 Stereopsis occurs when each eye sees a slightly different image [Media Lab 1998].

4.2 Interface components

The 3D user interfaces of today often take little advantage of the opportunities of the added third dimension using mostly plain 2D interface components that are mapped to rotation and translation of the 3D objects. A normal computer mouse is also commonly used. The mouse movement is then mapped to rotation/translation and a selection mechanism is used for selecting which degrees-of-freedom the device is controlling, as can be seen in figure 4-4.

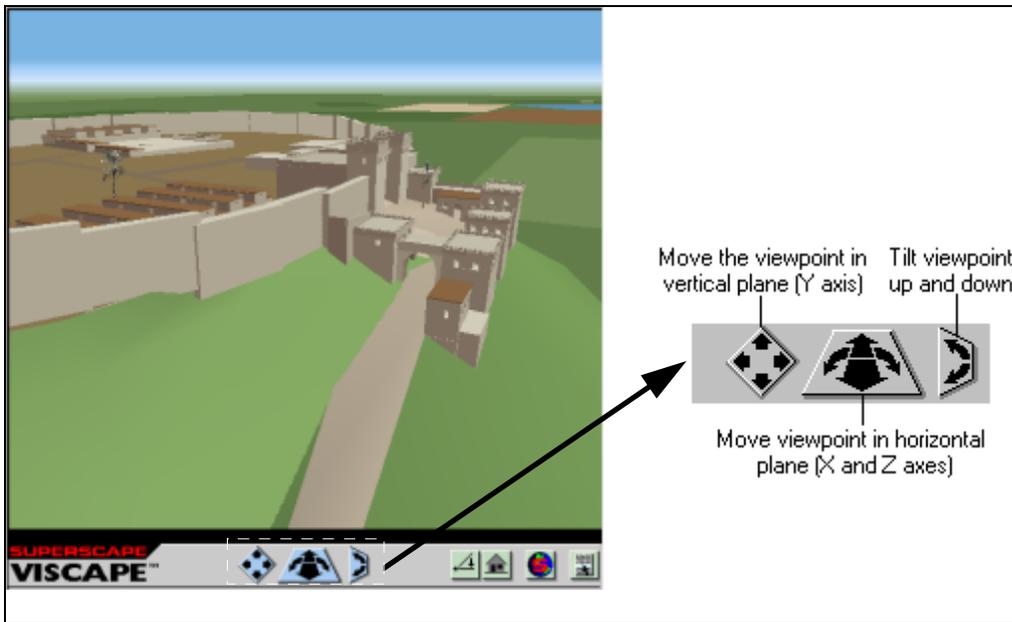


Figure 4-4 An example of 2D Interface components controlling the view of a 3D world [Superscape 1998].

Various approaches have been made to develop new metaphors and interaction techniques to take advantage of the greater possibilities of 3D. One approach that is still in its infancy is the concept of 3D widgets, which involves making the interface components first class objects in the same 3D environment as the rest of the objects in an application. There are no generally accepted metaphors for 3D interfaces analogous to the well-established metaphors for 2D interfaces discussed in section 3.2.1. The lack of metaphors denotes that essentially no toolkits for 3D interfaces exists [Conner et. al].

Selective Dynamic Manipulation is a paradigm for interacting with objects in visualisations [Chuah et. al 1995]. Users control the visualisations using 3D handles that may be attached to objects (see figure 4-5). To manipulate the objects the handles are pushed or pulled. The objects move and react continuously to the manipulation of the handles. Use of interactive animation in this way can help users understand and track the changes that occur to an object set [Chuah et. al 1995].

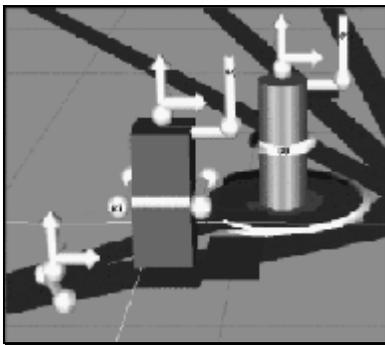


Figure 4-5 Selective Dynamic Manipulation uses handles on objects to afford stretching, rotation etc [Chuah et. al 1995].

Other examples of 3D widgets include the Arcball which is a mouse-driven 2D interface to simulate a physical trackball (see figure 4-6). The virtual object is shown on the screen, and when the user clicks and drags on the virtual object, the computer interprets these drags as tugging on the simulated trackball. The virtual object rotates correspondingly. To provide the third rotational degree of freedom, a circle is drawn around the object, and when the user clicks and drags in the area outside of the circle, rotation is constrained to be about the axis perpendicular to the computer screen [Hinckley 1996].

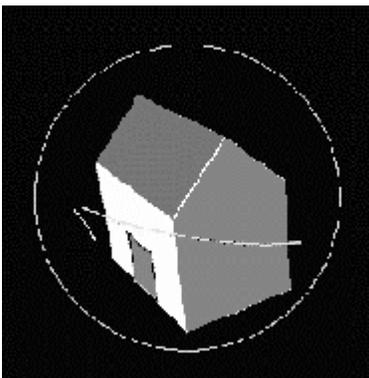


Figure 4-6 Example of the Arcball [Hinckley 1996].

4.3 Input Devices

Systems that use plain 2D widgets as interface objects require no special input devices besides a normal mouse and/or keyboard. The problems with using a normal mouse to manipulate 3D objects include that you have to indicate in some way which degrees of freedom you wish to control by moving the mouse. Other input devices are available to eliminate the need for multiple mouse clicks and menu selections. These can be divided in two categories: desktop devices and free-space input devices. Examples of desktop devices that provide six degrees of freedom are the Space Mouse and the Spaceball (see figure 4-7). These kinds of devices are often used as a complement to the 2D mouse to rotate objects or views. The 2D mouse is usually held in the dominant hand and is used for picking objects and selecting menu items while the other hand controls the 3D input device. In the case of the Spaceball input is accomplished by light twisting, pulling or pushing of the ball in all directions.

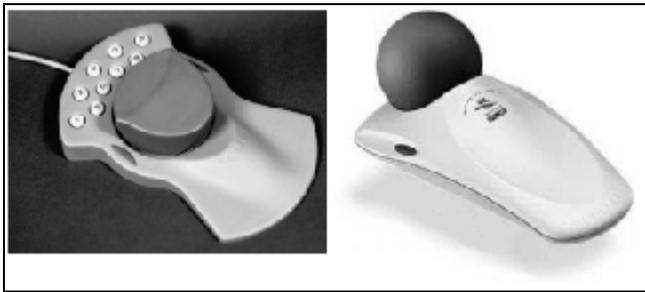


Figure 4-7 The Space Mouse (left picture) and the Spaceball are desktop devices with six degrees of freedom suitable for interaction with 3D objects.

The free-space input devices consists of camera-based devices and devices with magnetic trackers. Many different input devices using magnetic trackers are shaped as gloves, sometimes referred to as data gloves. The glove measures the movements of the wearer's fingers and transmits them to the computer. Sophisticated data gloves also measure movement of the wrist and elbow. A data glove may also contain control buttons or act as an output device, e.g. vibrating under control of the computer. An application usually generates an image of the data glove on the computer screen so that the user can grab objects in the computer generated 3D environment. By using different gestures such as pointing the user can change view or move around in the computer model.

A typical so called data glove is the Fakespace Pinch Glove (see figure 4-8) described in [Hinckley 1996]. The pinch glove uses tabs on the fingertips to sense when the user is pinching. On the back of the hand there is a magnetic tracker used to provide hand position data to the computer.

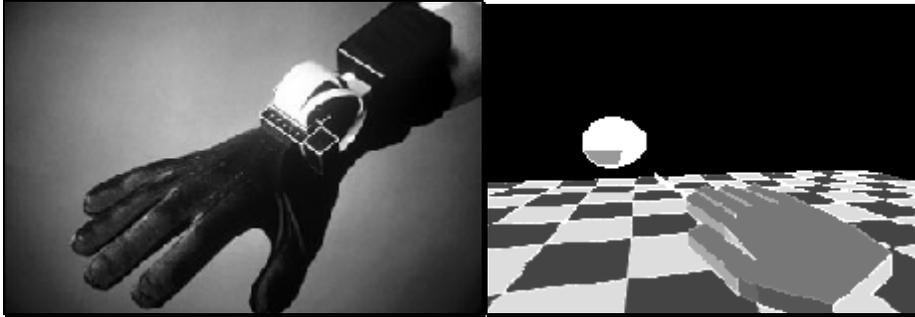


Figure 4-8 A data glove and its representation in the computer generated environment [Hinckley 1996].

4.4 Output Devices

3D output devices come in two types: monoscopic, in which both eyes see exactly the same view, and stereoscopic, where the eyes see separately computed views to give the sensation of stereoscopic vision [Preece et al. 1994]. Of course most 3D-applications use monoscopic output devices i.e. conventional computer monitors to display the 3D world. These applications are said to provide a “window on the world” interface. This is the technique with the least demands for computing power and special equipment but since no binocular depth cues are provided the users do not get a very strong feeling of depth.

The stereoscopic output devices include Head Mounted Displays (HMD) (see figure 4-9). HMDs are commonly used in the field of VR to create an immersive experience. A typical HMD is composed of a set of goggles or a helmet with tiny monitors in front of each eye to generate images seen by the wearer as three-dimensional. Often the HMD is combined with a head tracker so that the images displayed change as the head moves.

Other simpler forms of goggles for stereoscopic output are so called shutter glasses. They are used in combination with a normal computer monitor. The basic technique behind shutter glasses is that they alternately block eye views in synchrony with the computer display of left and right eye images to provide stereoscopic images on the computer screen. This technique cannot be used successfully together with head tracking since the computer monitor will not follow the user. Also shutter risk producing a flickering image.



Figure 4-9 The Head Mounted Display MRG 3c [Liquid Image].

A quite different approach to 3D output is the CAVE Automatic Virtual Environment from the University of Illinois. The CAVE is a surround-screen, surround-sound, projection-based

system The computer graphics are projected into a 10'x10'x9' cube composed of display screens that completely surround the viewer. The viewer explores the virtual world by moving around inside the cube and grabbing objects with a three-button, wand-like device.

Another way of providing true 3D output is the boom display described in [Hinckley 1996] (see figure 4-10).



Figure 4-10 The Boom Display.

Although the technology for HMDs is available today we consider it still to be in its infancy and it is not yet commonly used. It is hard to imagine an office of today using HMDs for common tasks. HMDs would significantly complicate ordinary communication or even such a simple thing as viewing what is located at your desk.

4.5 Virtual Reality

There are as many definitions of VR as there are people who try to define it. Although no single definition is available, a common goal of VR applications is to give the user an illusion of immersion in a computer-generated environment. The idea of VR originates from the books by William Gibson, where he describes a computer generated environment called “the Matrix” [Gibson 1984].

The term is sometimes used more generally to refer to any virtual world represented in a computer, even if it is just a text-based or graphical representation.

A more narrow definition of VR is sometimes referred to as immersive VR. It includes computer hardware that replaces some or all of the human senses. This creates an illusion of immersion in the virtual environment (see figure 4-11), e.g. replaces sight by using a Head Mounted Display, hearing by using headphones or the sense of touch or force on the body by using a special suit or gloves. For an application to fit in this definition of VR some or all of these techniques have to be included and combined.

When provided with techniques from the area of immersive VR we believe there is no need for widgets for translation and rotation of objects. If the user wants to rotate an object in such an environment it should be possible to grab the object in some way and rotate it as if it were an actual object.

Desktop VR is a subset of VR that does not require as much special hardware. It can be accomplished at little additional cost to a typical personal computer. The desktop VR applications all have a “window on the world” interface, they display the 3D world as if you were watching it through a window.



Figure 4-11 An immersive VR system using a head mounted display and a freespace input device.

The negative part is that the media has brought forward unrealistic expectations on the capabilities of VR. No computer application so far has provided the realistic feedback that purists claim to be necessary for VR. We believe that the real power of VR lies in the way object properties and behaviours are an approximation of their real life counterparts. Such features of VR make the objects' behaviour more accessible and understandable to human users. According to [Corder 1995], immersive VR is still in an experimental state but desktop VR and VR in the field of e.g. flight simulators already do a lot of good.

5 Application areas for 3D Graphical User Interfaces

This section covers examples of application areas where 3D GUIs have been applied. Examples of existing applications are given within the areas of information visualisation, simulation, navigation, entertainment, and marketing.

5.1 Information Visualisation

Systems based on the desktop metaphor use spatial metaphors but do not take advantage of them since they really are flat environments. The advent of 3D metaphors like rooms, houses, cities, etc. introduces the possibility of taking advantage of our spatial abilities like we are used to in the real world. Spatial concepts are undoubtedly very important ingredients in our daily lives and such spatial environments could therefore be of advantage in computer interfaces. This is shown in the File System Navigator described in [Andrews 1994].

The File System Navigator (FSN) from Silicon Graphics is a prototype known from its appearance in the movie Jurassic Park. The prototype visualises a Unix file system by displaying the files as an information landscape in an attempt to draw advantage of the users spatial capabilities (See figure 5-1).

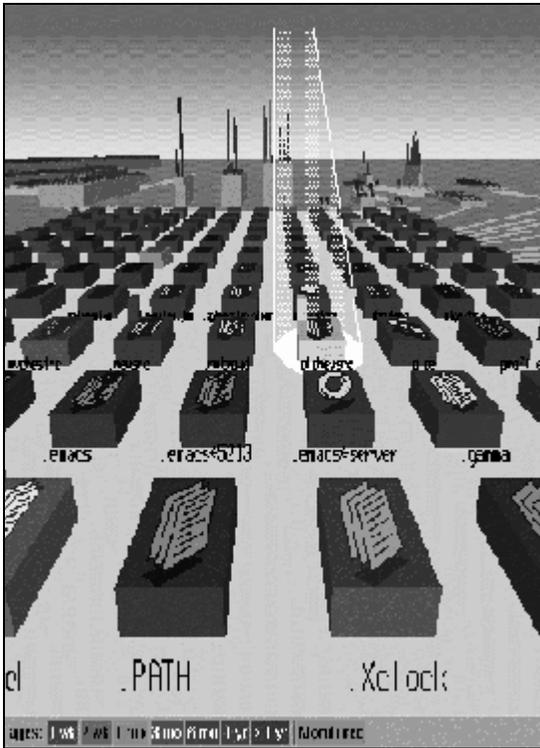


Figure 5-1 The File system Navigator visualises a Unix file system as an information landscape.

It uses 2D interface components but visualises the file system in 3D. Directories are represented hierarchically by pedestals laid out on a plane, their height is proportional to the size of the contained files and different colours represent the age. Smaller blocks on top of the directory pedestals represent files in the directory. The directories are connected by wires, on which it is possible to travel. Clicking on the arc to a subdirectory results in an invisible hand grabbing you and leading you through space to that subdirectory. Clicking on a file block

brings a virtual spotlight to mark the box and double clicking opens the file for editing. Whether this actually helps the user to build up a good mental representation remains to be seen, but FSN certainly gives an interesting perspective on well-known directory trees such as the Windows New Technology (NT) Explorer.

5.2 Simulation

A common goal of applications like flight simulators or driving simulators is to provide an artificial environment that looks, feels, and reacts like its corresponding real life environment. In these applications, 3D GUIs are used to create an immersive and realistic experience for the user.

5.2.1 Virtual Wind Tunnel

The Virtual Wind Tunnel from NASA allows scientists to visualise properties of airflow and pressure in computational fluid dynamics volumetric data sets. [Corder 1995]

The virtual wind tunnel supports a variety of visualisation techniques. Most of these techniques are designed to allow the researcher to move the visualisations around, exploring phenomena in a desired region of space. In order to allow such direct manipulation of the visualisations, each visualisation technique must somehow be associated with a point in space. Then the location of this point is used to determine what data that visualisation is displaying.

There are many aspects of the virtual wind tunnel, which do not lend themselves to a direct manipulation control paradigm. These include assignment of data, control of the number of emitters, and the creation and deletion of the visualisation control tools. Control over aspects like these is provided by embedding a simple generalisation of conventional GUI techniques into the virtual wind tunnel.

Interaction with the data is achieved with a glove for pointing and issuing commands by gesturing plus a boom display for viewing the virtual environment (See figure 5-2). The boom display is used instead of a head-mounted display to avoid problems with encumbrance. The boom also makes it easy for a pair of scientists to share views of the data.

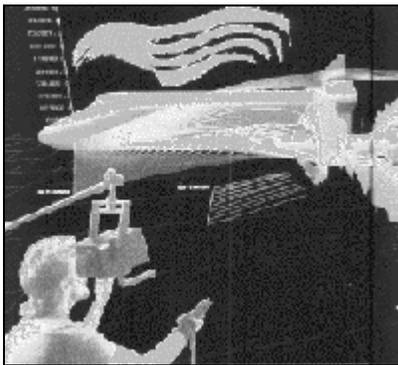


Figure 5-2 Using a boom display and glove with the Virtual Wind Tunnel.

5.2.2 Flight Simulator

Flight simulators allow pilots to become familiar with new aircraft before piloting them in the real world. It also allows experienced pilots to practice advanced procedures that might be too dangerous in real life. The more advanced flight simulators were originally only used for military and commercial flight instruction purposes.

The Vertical Motion Simulator from NASA is one of the largest three-dimensional simulators in the world [Corder 1995]. It consists of a simulated air craft cockpit sitting on a platform which is itself supported by a number of hydraulic pistons to move the whole cockpit around (See figure 5-3).

The simulated cockpit consists of a removable room that is used to represent different types of aircraft such as fighters, helicopters, transports, and space shuttles. The cockpits differ in the number of windows, configuration of the controls and instruments, since they must be able to represent a wide variety of current and planned aircraft. Behind the windshields are high-resolution displays designed to approximate what the pilot would see in the corresponding real world situation.

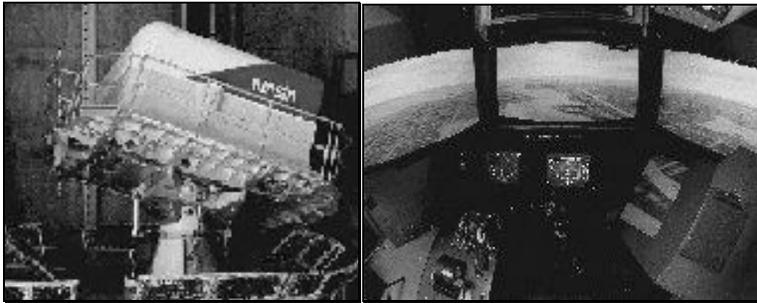


Figure 5-3 The simulated air craft cockpit of the Vertical Motion Simulator.

5.3 Computer Aided Design

Computer Aided Design systems are designed to facilitate the design process for engineers and architects when creating and manipulating 2D and 3D models. These models may be rotated, scaled to various proportions, edited, labelled, and manipulated in a variety of ways. CAD systems deal with displaying and manipulating blue prints for real objects and can therefore benefit from 3D graphics to a great extent (see figure 5-4).

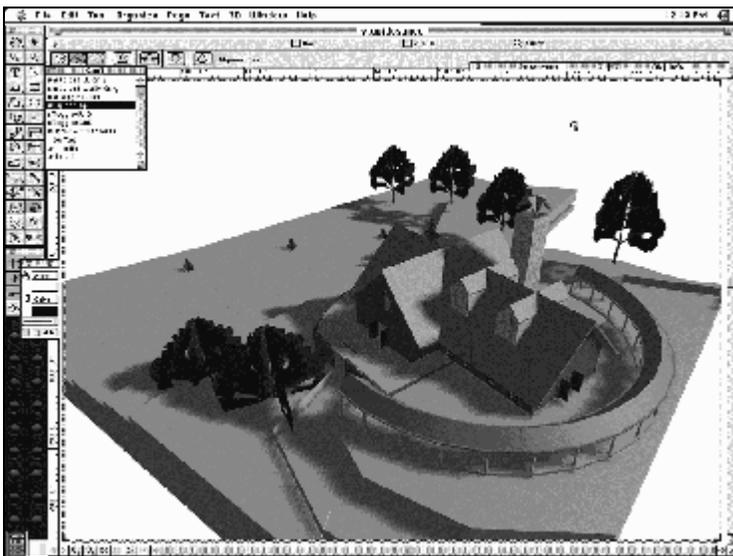


Figure 5-4 The CAD system MiniCAD for the Apple Machintosh.

The 3D models created with the CAD system can be used to create architectural walk-throughs. Interactive computer programs that simulate the experience of “walking” through a buildings interior are useful for visualisation and evaluation of building models before the buildings are constructed. With a walk-through, things can be discovered, added or corrected before the actual construction begins, thus potentially saving a lot of money.

5.4 Computer Supported Cooperative Work

Computer Supported Cooperative Work (CSCW) is the study of how people work together using computer technology. Typical applications include email, awareness and notification systems, videoconferencing, chat systems, multi-player games, and real-time shared applications (such as collaborative writing or drawing). The potential benefits of designing computers and their interfaces to facilitate group working appear to be huge [Preece et al. 1994]. As shown in the next section virtual environments can provide powerful and natural ways of collaboration using computers.

The Distributed Interactive Virtual Environment (DIVE) system from the Swedish Institute of Computer Science (SICS), is a toolkit for building distributed multi-user VR applications in a heterogeneous network environment [Stål and Andersson 1997]. As shown in figure 5-5, participants navigate in 3D space and observe, meet and collaborate with other participants. A participant in the world is called an actor, and is either a human user or an automated application process. To facilitate the recognition and awareness of ongoing activities, the actor is represented by an avatar, which is an icon or a representation of a user in a shared virtual reality world or multi-user domain [Carey and Bell 1997].



Figure 5-5 Conference in the DIVE.

The visualiser is a rendering² application through which the user sees the world, it renders the scenes from the viewpoint of the actor’s eye. Changing the position of the eye, or changing the eye to an another object, will change the viewpoint. A visualiser can be used in combination with a wide range of devices such as an HMD and gloves. Further, it reads the user’s input devices and maps the physical actions taken by the user to logical actions, such as navigation and interaction, with the objects in the Dive system.

² Rendering is the process by which a computer creates images from models [Nieder et al. 1993]

In a typical DIVE world, a number of actors leave and enter dynamically. Additionally, any number of applications exists within a world. Such applications typically build their user interfaces by creating and introducing necessary graphical objects. Thereafter, they scan after events in the world, so when an event occurs, the application reacts according to some control logic.

5.5 Entertainment

Whether you look at computer games or movie effects there is no doubt that 3D modelling has had a tremendous impact on the entertainment industry over the last decade or so.

Many movies of today feature special effects in 3D and may even become blockbusters just because of them. As described in [Govil-Pai 1998], there even are movies completely made up of computer animation e.g. Toy Story. Most new computer games include 3D graphics and 3D user interfaces in one form or another. This implies that users prefer the games with 3D graphics to their 2D counterparts. One possible reason is that new computer games must look modern to attract its audience. Many game developers have spent a lot of money and effort developing attractive, amusing, and easy to use 3D interfaces. According to us, this suggests that there is much to learn from studying the user interfaces of these games.

To apply techniques from games in e.g. office applications, one must bear in mind that computer games do not necessarily have to include a user-friendly interface. A problem in controlling the new user interfaces can be viewed simply as part of the game.

5.6 Marketing

The computer, telephone, and television boom, as well as the merging of these technologies, has had a major impact on the way companies produce and market their products. The advent of the Internet promises companies access to millions of new customers at a fraction of the cost of printed or televised advertising [Kotler 1997]. With relative ease multimedia documents can be constructed for the WWW. Text and pictures can provide a great deal of product information, but are static and limited to what the creator chooses to present.

It would be more useful if a potential buyer was able to interact with an animated 3D model of the product. This would engage the attention of the buyer, and could provide information of the product that otherwise could only be gained from real experience of the product [Burton et al. 1997]. Therefore, it is important that the technology for marketing a product is able to attract and maintain the attention of a potential customer. It must be easy to use even for inexperienced users, it should also provide useful information about the product. Finally, it must be cost-effective to employ and easy to create and maintain.

People do find 3D interesting and if there is sufficient 3D interaction built into the environment, it will not only attract interest but maintain it as well [Burton et al. 1997], thus making the users in control not the computer. As we have mentioned before, the main problem with this technology is the lack of developed 3D navigation techniques. Other drawbacks are the time it takes to create complex scenes and the slow rendering of these scenes.

There are basically two different ways of using 3D graphics for marketing purposes. Attractiveness, i.e. attracting users by making applications look and feel and advertising. Advertising possibilities include making impressive commercials for television and the cinema using computer graphics, but there is also a new way of advertising possible, namely the Internet. Using modelling software, companies can create 3D models of their products. This allows an interactive form of marketing where the customers have more possibilities of examining a product before is bought.

6 Overview of 3D Graphics Standards

The use of 3D graphics has been limited not only by the performance of computer systems but also by the lack of programming standards. Without standards, it has been a big undertaking for developers to include 3D interfaces in their applications.

Standards allow developers to include 3D graphics in their applications without having to “reinvent the wheel” over and over again. Standards also allow for hardware developers to optimise their hardware for the use of that particular standard. This leads to higher performance and hopefully also better cross-platform compatibility, making it easier to port applications to other platforms.

This chapter will focus on two standards; Open Graphics Library (OpenGL), a low-level programming interface for 3D graphics and Virtual Reality Modelling Language (VRML) a high-level description language of 3D worlds. These standards are well accepted at the writing of this thesis and they represent two different approaches to standardisation of 3D graphics. The insights gained from writing this chapter guided us into the choice of tools for the implementation of the prototype described in chapter 8.

6.1 OpenGL

This chapter provides an overview of OpenGL with focus on the design considerations in the design of the system. In the end of the section a list of application areas for OpenGL is presented.

6.1.1 Introduction

OpenGL is a software interface to graphics hardware designed to be independent of operating system, window system, and hardware operations. The interface consists of commands that allow specification of geometric objects in two or three dimensions, together with commands that control how these objects are rendered. OpenGL has no high-level commands for describing complex 3D objects, instead it includes a set of geometric primitives—points, lines, and polygons which when used in combination can build up more complex models (see figure 6-1).

6.1.2 History

Silicon Graphics Inc. developed OpenGL as a successor to the graphics library called IRIS Graphics Library (IRIS GL). IRIS GL is a hardware independent graphics interface that has been implemented on numerous graphics devices of varying sophistication. [Kilgard 1993]

OpenGL has a 3D rendering philosophy similar to IRIS GL, but has removed outdated functionality or replaced it with more general functionality making it a distinctly new interface. Experience with IRIS GL was useful in designing OpenGL and another advantage was the fact that programmers with experience of IRIS GL were able to change systems without great trouble.

In 1992 OpenGL was proposed as a standard for 3D graphics and for this reason the industry consortium known as the OpenGL Architectural Review Board (ARB) was formed. The OpenGL ARB licenses and directs further development of OpenGL. Current members are

Digital Equipment, Evans & Sutherland, Hewlett-Packard, IBM, Intel, Intergraph, Microsoft, Silicon Graphics, and Sun Microsystems³.

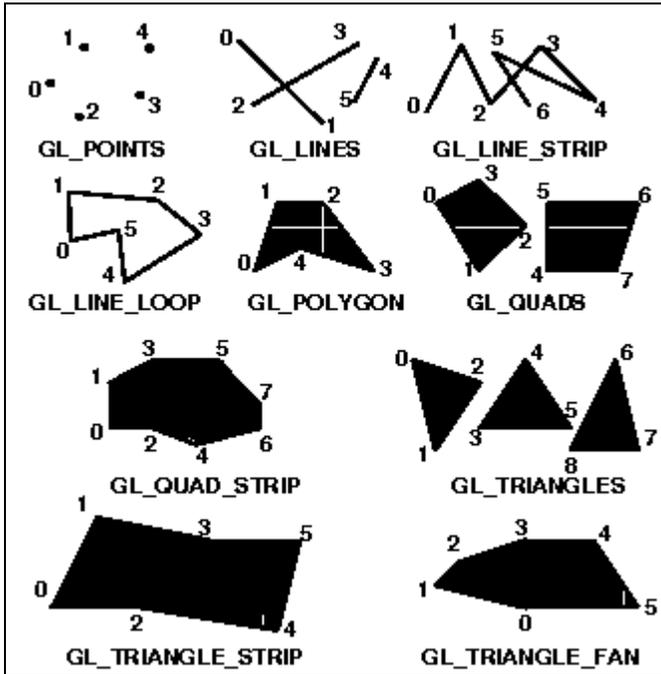


Figure 6-1 OpenGL Geometric Primitives.

6.1.3 Design

OpenGL provides a layer of abstraction between graphics hardware and an application program. To the programmer it is visible as an application programming interface (API) consisting of about 120 distinct commands. For an example see figure 6-2.

```

glBegin(GL_POLYGON);
    glVertex2f(0,0);
    glVertex2f(0,1);
    glVertex2f(1,0);
    glVertex2f(1,1);
glEnd();

```



Figure 6-2 OpenGL code used to describe a rectangle.

The specification is window system independent meaning it does not specify how to manipulate windows or receive input from the user. This allows OpenGL to be implemented for any window system.

³ According to the OpenGL ARB FAQ (<http://www.opengl.org>)

6.1.4 Architecture

To maintain good performance rates the OpenGL API allows complete access to graphics operations at the lowest possible level that still provides device independence. As a result it does not provide a means for describing or modelling complex geometric objects [Segal 1994]. This means that the OpenGL commands specify how a certain result should be produced rather than what exactly that result should look like, i.e. OpenGL is procedural rather than descriptive.

OpenGL uses immediate mode rendering. That is, when the graphics system is used to create a scene or object, each function and command has an immediate effect on the frame buffer and the result of each action is immediately visible on the scene.

The boundary between the OpenGL implementation and an application is well defined to clearly specify how data is passed between the two. An application (the client) issues commands that are interpreted and processed by OpenGL (the server). However the client-server model is an abstract model and does not demand OpenGL to be implemented as distinct client and server processes. If it is run on a single computer, that computer is both client and server.

6.1.5 Application Programming Interface Specification

The designers of OpenGL present the graphic system as a state machine. Specifying the API as a state machine allows consistent, precise specification and eliminates ambiguity about what a given operation does and does not do [Kilgard 1993]. The routines that OpenGL supplies provide a means for the programmer to manipulate OpenGL's state machine to generate the desired graphics output. The programmer puts the machine in different states/modes that remain in effect until he changes them. The programmer can at any time query the system for the current value of any of the state variables. E.g. one of the variables holds the current drawing colour: after it is set, every object drawn will have that colour until the variable is changed. [Neider et al. 1993].

To ensure that all implementations of OpenGL fully conforms to the standard, all implementers must perform a suite of tests to ensure source code compatibility [Neider et al. 1993]. This allows users/programmers to be sure all basic features and functions will be available on all different platforms.

The OpenGL API is defined in terms of the programming language C, but bindings for several other languages exist. Fortunately the OpenGL function parameters are all simple types (Boolean, integer, floating point, arrays) so the API translates easily from C to other languages. Currently the ARB controls C, C++, Fortran, Pascal, and Ada binding specifications. However several unofficial bindings exists for other languages, e.g. Java, Tcl/Tk, and Python. [Neider et al. 1993].

6.1.6 Libraries on top of OpenGL

Higher-level drawing is cumbersome using only OpenGL's powerful but primitive set of drawing commands. This and the lack of support for communication with a specific window system have led to the development of several libraries written on top of OpenGL, e.g. OpenGL Utility Library, OpenGL extension to the X window system, and Open Inventor.

The OpenGL Utility Library

The OpenGL standard also defines the OpenGL Utility Library (GLU). The GLU serves both as a means to render useful geometric objects and as a model for building other libraries that use OpenGL for rendering. Among other things the GLU includes routines for simplifying rendering of spheres, discs, and cylinders.

OpenGL Extension to the X Window System

The X window system has become standard for UNIX workstations [Segal 1994]. The programmer uses the window system to obtain windows on the display on which he can display images or text. The window system also provides the programmer with means to obtain user input via keyboard or mouse. OpenGL Extension to the X Windows System makes OpenGL rendering available to the programmer just like any other extension to the window system.

Open Inventor

Open Inventor represents an object-oriented application policy built on top of OpenGL, providing a programming model and user interface for OpenGL programs [Wernecke 1994]. Open Inventor provides pre-built objects and a built in event model for user interaction, high-level application components for creating and editing three-dimensional scenes, and the ability to print objects and exchange data in other graphics formats. Open Inventor also defines a file format for describing 3D scenes that were used as a base for the design of the Virtual Reality Modelling Language (see section 6.2).

6.1.7 Applications

OpenGL is according to us a platform well suited for building libraries that handle structured geometric objects, no matter how the structures are described. This is because of its independence of particular methods for describing the objects.

A common use of OpenGL is in the field of direct manipulation where the graphics engine has to perform fast rendering to provide immediate feedback to the user. In the making of the movie Jurassic Park an Open Inventor based lighting and shading editor was used. The editor provided the graphic artists with a tool that could quickly show the effects of changing lights [Wernecke 1994]. Other applications of OpenGL include CAD/CAM applications, scientific visualisations, and games.

6.2 Virtual Reality Modelling Language

This section is an introduction to the history and basics of the Virtual Reality Modelling Language (VRML). The languages' current problems and the future of the language are also discussed.

6.2.1 Introduction

VRML is a simple platform independent language for describing different 3D shapes and interactive environments. VRML can be used in a variety of areas, e.g., engineering, scientific visualisation, multimedia presentations, entertainment, educational titles, web pages, and shared virtual worlds [Marrin et al 1997].

6.2.2 History

In 1989 a new project was started at Silicon Graphics, Inc., with the aim to design and build an infrastructure for interactive 3D graphics applications. In 1992 the Iris Inventor 3D toolkit was released as the first product of these efforts. In 1994 a major revision of Inventor was released under the name Open Inventor. Open Inventor is based on OpenGL and is portable to a variety of platforms. The reference manual describing the objects and file format in the Open Inventor was eventually used to write the first draft of the VRML1.0 specification. The second release of VRML adds significantly more interactive capabilities to the language [Carey and Bell 1997]. The formal processing of the current version called VRML97 to become an International Standard began in June 1996.

The VRML Consortium, Inc. is a non-profit corporation that was formed in 1996 with the mission of fostering and promoting VRML as the open standard for 3D multimedia and shared virtual worlds on the Internet. Today the Consortium has 52 Members including among others Apple computer, Silicon Graphics, Fujitsu, IBM, Intel, Microsoft, Mitsubishi, Oracle, Samsung, and Sony [Marrin et al 1997].

6.2.3 Design

The design of the language is influenced by several general design philosophies. These philosophies have arisen from the constraint put on the design by the VRML standardisation process and the intended users of VRML [Carey and Bell 1997]. VRML is a high-level language designed to fit into the existing infrastructure of the Internet [Marrin et al 1997]. Using existing standards makes it easier for the Web developer as well as for the implementers of the VRML standard. VRML can contain references to files in many different graphical and sound standard formats as well as programming languages such as Java [Carey and Bell 1997]. VRML allows graphic designers to create objects in a 3D space and to add simple behaviour to those objects. Java can be used to allow more complex behaviour or provide access to external data. There are basically two ways to use Java in combination with VRML. It can either be used as a scripting language inside the VRML code, or as an external Java applet that controls a VRML world (See section 6.2.5) [Carey and Bell 1997].

VRML can be regarded as a 3D counterpart to the World Wide Web standard HyperText Markup Language (HTML) [Carey and Bell 1997]. This means that VRML serves as a simple multiplatform language for publishing 3D Web pages. One important shared property of VRML and HTML is the simple specification and the ease to take files created by different people or tools and merge them into a new document (see figure 6-3).

```
#VRML V2.0 utf8
Shape {
  Appearance
  Appearance {
    material Material { }
  }
  geometry Sphere {
    radius 1.0
  }
}
```

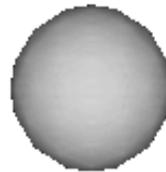


Figure 6-3 Example of VRML source code to create a sphere.

6.2.4 Presentation and Interaction

All the interpretation, execution, and presentation of VRML files is done by a browser, which displays the scene graph [Carey and Bell 1997]. The scene is an acyclic directed graph, which contains hierarchically grouped nodes that describe the objects and their properties [Marrin et al 1997]. This presentation is known as a virtual world and is navigated by a user. The world is displayed as if experienced from a particular location. The browser may define navigation paradigms that enable the user to walk or fly through the virtual world.

The user can also interact with the world through sensor nodes in the scene graph. Sensors respond to user interaction through the different objects in the world, the passage of time, or the movement of the user through the world.

The visual presentation of the objects in the virtual world is designed to resemble the physical characteristics of light. A lighting model is used to describe how appearance properties and lights in the world are combined to produce the displayed colours [Carey and Bell 1997].

The browser consists of three main components (see figure 6-4), the parser, the scene graph, and the audio and visual presentation [Marrin et al 1997]. The parser interprets the VRML file and creates a scene graph. The scene graph consists of a transform hierarchy (the nodes), and routes graph (the connections between the nodes). There is also an execution engine in the scene graph that processes the route graph, and makes changes to the transformation hierarchy. User interaction is transferred to the route graph component and the audio and visual presentation. The audio and visual presentation performs the graphics and audio rendering of the transform hierarchy and feeds it back to the user [Marrin et al 1997].

6.2.5 External Authoring Interface

The External authoring interface (EAI) is a standard programming interface that defines the set of functionality of a VRML-browser that the external environment e.g. a Java applet can access [Marrin 1997]. It allows users to write external programs that interact with and dynamically updates a world. Such VRML interfaces could for instance be used to help organise complex data sets in real time. This allows VRML to be part of a larger multimedia web presentation.

Several VRML-browsers support EAI, which is not yet a part of the VRML specification [Marrin et al 1997]

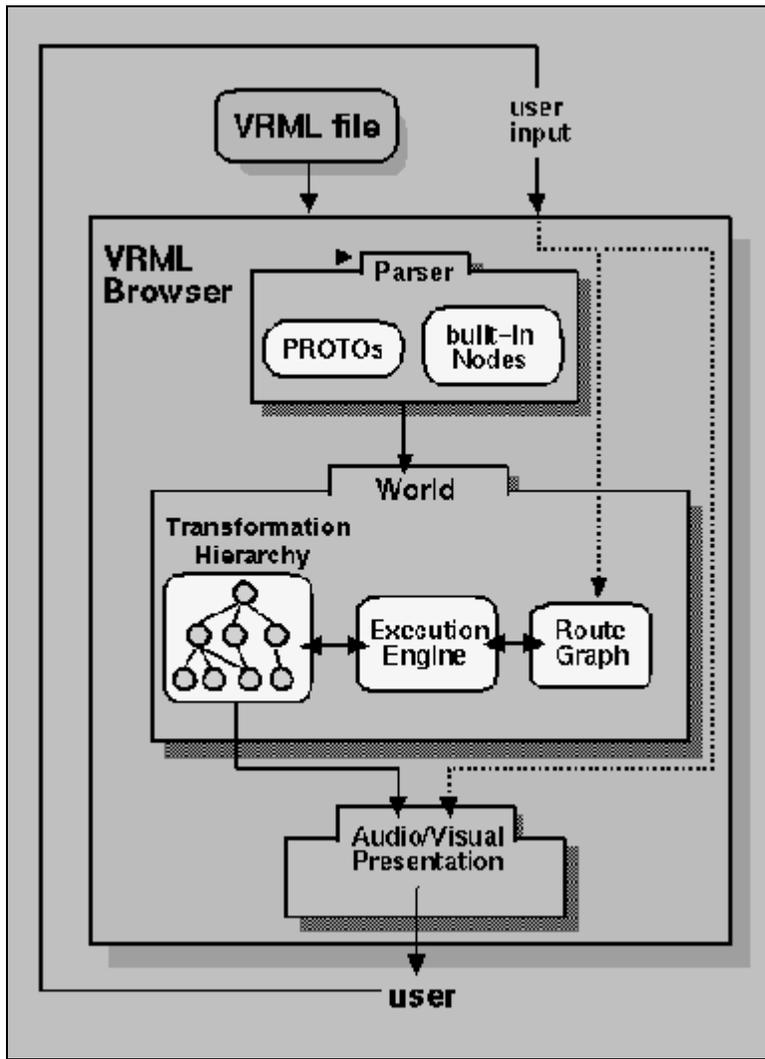


Figure 6-4 A Conceptual Model of a VRML Browser [Marrin et al 1997].

6.2.6 Problems

The language has the same types of problems as HTML. There are browsers for most of the common platforms, (Macintosh, Windows NT/95/3.1, SGI Irix, SUN) but unfortunately the browsers are not all consistent with the VRML specification nor with each other. Some of the major problems encountered according to [Sanders and Gay 1996] are:

- Different browsers can display the same object with different colours and shading.
- Worlds written in a given programming language will only run on browsers that support that particular language.
- Most browsers let the users interact with the world while it loads, this means that the developers must design user interaction with caution to prevent worlds from crashing if only partially loaded.
- One of the major problems with the language is that it is new; neither browsers, nor tools are completely up to the final specification.

Other problems that we have encountered are that many complex objects with texture mapping makes the VRML worlds slow and sometimes almost impossible to explore. It is

also difficult to create source code that is easy to read and follow thus making it difficult to understand or debug the source code. Complex VRML environments are therefore most likely to be created with some kind of authoring tool.

6.2.7 Future

VRML is a novelty that we feel has potential, but not many essential applications today. It is at a relatively early stage of development and promises to be a prominent technology in the future. It has received industry-wide support as a standard to build on for the future of the web and is unique in that it does not have much competition.

There are several VRML extensions in progress. Many users and developers have requested a compressed binary format for VRML files, emphasising the reduction of file size for fast transmission, and also a simplification of the file structures for fast parsing [Carey and Bell 1997]. There has also been a request for a standard programming interface for VRML browsers to be defined. Finally, many feel that a very important issue is about adding multi-user interaction or autonomous creatures (i.e. avatars) that can sense and react to their environment [Carey et al. 1997].

6.3 Conclusion

OpenGL and VRML are two different standards that are good at different things. Using OpenGL requires a lot of relatively low-level programming. VRML on the other hand uses a more high-level approach similar to HTML thus making it simple to produce 3D models. When creating simple models, that are to be published on the Internet e.g. as part of a web site, VRML is to prefer. When creating more complex applications using 3D graphics OpenGL is the better choice.

Other potential standards for 3D graphics have emerged but have not reached the same amount of spread as VRML or OpenGL. To show the reader that there are alternatives to the earlier described standards, this section gives a quick overview of other potential standards for 3D graphics.

Direct3D is a Windows API developed by Microsoft offering similar functionality to OpenGL. This API is used primarily to provide 3D acceleration in games.

Java3D is an extension to the Java programming language. With Java3D the programmer can create scene graphs similar to those in VRML.

Quickdraw 3D is a cross-platform API from Apple. It also includes a set of interface standards and guidelines to promote consistency between applications. [Young 1996]

Superscape Virtual Reality Toolkit (VRT) is a commercially available virtual reality authoring system from Superscape based on the PC platform. The VRT consists of several different editors combined into a single authoring environment. A plug-in and an ActiveX control allow you to navigate within 3-D worlds on the Web.

7 Comparison of 2D and 3D Graphical User Interfaces

We do not intend to argue that any of the 2D or the 3D GUIs is better than the other. Each interface device or input technique will work better for some tasks and worse for others. The most appropriate user interface for an application depends on the context of tasks to be supported and the intended users.

The discussion in this chapter is based on the cognitive aspects of user interface design described in chapter 2.

7.1 Navigation

One could argue that since we live in a 3D environment in real life, this would be the more natural way to navigate computer systems. On the other hand it has been shown that compared to 2D GUIs, 3D GUIs provide new ways to get lost. In this section we discuss navigational aspects of 2D and 3D GUIs i.e. cognitive maps and spatial abilities. The problem with motion sickness in 3D GUIs is also discussed.

7.1.1 Cognitive Maps

A problem can be noticed while exploring 2D hypermedia environments, like the Internet. The space is not fully known so users must construct their own maps, using search engines or browsing, easily confusing the navigator [Jul and Furnas 1997].

Since we live in a 3D environment in real life, it is possible that it could be more convenient for us to navigate in the same way when we use our computers. Compared to a 2D GUI a 3D GUI more resembles a real world environment. The 3D GUI can e.g. include generic components like landmarks, paths etc. as discussed in section 2.1.1. This implies that one would be able to take more advantage of the users' natural ability to create a cognitive map, thus facilitating the users' navigation of the information space [Darken and Sibert 1993].

7.1.2 Spatial Ability

One of the major problems in 3D user interface design, according to [Boyd and Darken 1996] is that with the increased degrees of freedom, new and interesting ways to get lost while moving through the worlds are provided. As suggested by [Höök and Dahlbäck 1997] users' spatial ability affects their ability to make use of computer systems. The ability to perceive and understand spatial relationships is of great importance when using a 3D GUI. This implies that Höök and Dahlbäck's results are also applicable on these GUIs.

7.1.3 Motion Sickness

A problem similar to motion sickness has been experienced by users during, and sometimes after, a session in a 3D environment. Early studies with both fixed-base and moving-base driving simulators implicated cue conflict as the source of the problems [Goldberg and Hiller 1995]. The theory of cue conflict has been described in [Goldberg and Hiller 1995] to occur when there is an ambiguity between senses or within a sense. The two primary conflicts thought to be at the root of simulator sickness occur between the visual and vestibular senses. In a fixed-base simulator, the visual system senses motion while the vestibular system senses no motion. Thus, according to the cue conflict theory, a conflict results. In a moving-base simulator, the visual stimuli experienced may not correspond exactly to the motion that the vestibular system registers. Thus, the result can still be a conflict.

This problem needs to be solved before 3D graphics can replace 2D graphics as the standard user interface in everyday work.

7.2 Interaction

Since the breakthrough of the desktop metaphor most users have developed an understanding of 2D interface components. No such commonly accepted standards are available for 3D GUIs but 3D has other advantages e.g. its potential to take advantage of differences in visual appearance. This section compares cognitive aspects of interaction with 2D and 3D GUIs i.e. attention, affordance and abstraction. Aspects of input and output are also discussed.

7.2.1 Attention

The interface design should allow the users to focus attention with as little cognition overload as possible. [Preece et al. 1994]

Techniques for presenting information at the interface to guide attention in 2D GUIs has included the use of spatial and temporal cues, different colours, and alerting techniques such as blinking, pop-up boxes, and auditory warnings.

3D GUIs can take better advantage of the differences in visual appearances that attract human attention, such as colour, shape, texture, shading, size, location, and the ability to register movement [Wiss and Carr 1998]. A drawback is that if used wrong, size can be confusing, since size is a method to create a feeling of perspective in 2D display. Another drawback with 3D GUIs is that objects in the interface can stand in the way of each other when viewed from certain angles.

7.2.2 Affordance

The direct manipulation paradigm has caused the development of a kind of “visual language” for affordances in applications. Starting with the Star interface, the desktop metaphor has been established as a standard throughout different platforms (e.g., Macintosh, Microsoft Windows) [Preece et al. 1994]. Therefore most users have acquired some sort of understanding of the affordances of general 2D interface components [Wiss and Carr 1998]. This is despite the fact that the GUIs of these operating systems no longer have a strong connection to the actual real world desktop.

The paradigms and metaphors of 3D interfaces are less developed than those for 2D interfaces, thus forcing the developers of 3D applications to develop new metaphors and paradigms or to utilise 2D interface components in a 3D interface [Conner et al. 1992]. Some 3D metaphors are natural analogies of those familiar in 2D, however developers of 3D interfaces must find new metaphors and interaction techniques to take advantage of 3D environments. The use of real-world metaphors could be used to support affordances, e.g., an object with the resemblance of a box can be opened. After all we live in a true 3D world and by the time we reach adulthood we have perfected many manipulative and navigational tasks to the point where we can perform them without conscious attention. This level of naturalness applied to a 3D user interface would be intuitive and easy to learn.

7.2.3 Abstraction

A technique often used when implementing 3D graphics is to reduce the level of details for objects far away from the viewpoint. The reason for this is the limitations of the computer hardware but it is also done in order to reduce cluttering. Only the nearest objects are displayed in full detail. Objects far away can e.g. be displayed as simple cubes that the user

perceives as abstractions of the actual objects. This technique can for example be used to let the user get a general view of a large amount of information.

Different levels of detail can also be used in 2D interfaces, e.g. for displaying maps with different magnifications, but it is even more useful when applied to 3D graphics.

7.2.4 Input and Output

2D interface components often have very simple behaviour. They commonly have few Degrees of Freedom (DOF) and usually support only a small range of values within that DOF. Interaction in a 3D space on the other hand deals with six degrees of freedom versus three in a 2D environment [Conner et al. 1992]. Users need to examine 3D objects from different viewpoints and navigate through scenes to gain a better understanding of the world. Therefore viewing and interacting with 3D data demands more user interaction. In our own informal studies we have not found any advantages in displaying blocks of text in three dimensions, since it only leads to the cumbersome task for the user to try to view the text in a straight angle from above.

Sufficient hardware in 2D desktop environments usually consists of a display, keyboard, and a mouse. The same kind of equipment can also be used for a 3D environment with various modes and mapping to compensate for the lack of degrees in freedom [Hand 1997]. To really take advantage of 3D technology, input and output devices that support interaction in 6-DOF are to be preferred.

2D graphics of today is well understood and the hardware has been optimised by the vendors. Most applications that present a 3D world still use the same kind of display as a normal 2D application, and to some extent also the interaction techniques of the desktop metaphor, disregarding the fact that this could be inappropriate for a large class of 3D applications. Almost all 3D-interaction techniques must be created from scratch since there are no essential toolkits specific for 3D interfaces. This places the burden of implementing interface components on the developer.

7.3 Discussion

As we stated earlier, “The most appropriate device for an application depends on the context of tasks to be supported and the intended users”. A more controversial standpoint in this matter could be to claim that 3D GUIs are like colour television. In the beginning days of colour television people were sceptical about it but nowadays almost no one will settle for a television set without colour display. Since it is more natural for us to watch a television show in colour since we can perceive colour, would it not be more natural for us to be able to rotate objects in all directions instead of just having to see them in two dimensions? One possibility is that all objects in a computer program always will have a 3D representation but that the user has the option of ignoring it. Compare it with a colour television set that can also be used to watch programs in only black and white. Imagine e.g. a computer program with a desktop metaphor where all objects have 3D representations. The user is able to view the desktop from any angle but chooses just to watch the paper he is writing from above.

8 Analysis

To illustrate how 3D graphics can be applied to user interfaces, we have developed a prototype with a GUI applicable to several of the existing products of ABB Network Partner. We call it A Bumble Bee.

In this chapter we will therefore describe the current and wanted situation for the user interfaces of the applications of ABB Network Partner. This will lead to a description of a new user interface.

8.1 Current Situation

Most applications of ABB Network Partner run in Windows or UNIX environments with traditional GUIs. The user interfaces of these applications are described in Appendix A.

One common element in a number of these applications is the visualisation of data in hierarchies using a tree view. These hierarchies can contain a substantial amount of information e.g. all information about relays in a complete power company like Vattenfall⁴. This leads to very large hierarchies that are difficult to take stock of.

Products like S.P.I.D.E.R. and IDES/RDE rely largely on the use of map displays in which power stations are placed. In the case of S.P.I.D.E.R. the map is used for surveillance purposes, while it is used as a graphical interface to a database in IDES/RDE. These maps must be easily navigated since they contain a lot of information that sometimes must be available fast in critical situations like power failure.

8.2 Desired Situation

This section gives some examples of how we believe that 3D interfaces can be used in the future to gain profit.

The rapid changes in technology of today force all companies involved in the business of information technology to keep track of the latest development. Companies see technology as producing an endless stream of opportunities. Yet taking advantage of technology entails walking a thin line; companies must avoid jumping in too soon (before the market is ready) or too late after the market has been conquered [Kotler 1997].

Expectations on 3D graphics that have emerged from within ABB Network partner include providing new ways of information visualisation and creating attractive, modern looking user interfaces. Another reason for ABB Network partner to use 3D graphics in their applications is to keep up with competitors also conducting research in this area.

Our description of the desired situation is based on informal interviews with representatives from the HCI-field and engineers within ABB Network Partner. Basically two common application areas emerged out of these interviews. This section is therefore divided into these two parts; information visualisation and marketing and can be seen as being a summary of expectations on 3D graphics of ABB Network Partner. Other possible application areas for 3D GUIs within ABB Network Partner, as described in chapter 5, are excluded.

⁴ Vattenfall is one of the largest power production companies in Sweden.

8.2.1 Information Visualisation

The increase of the computer availability over the last couple of years, in combination with advances in network technology, has resulted in some sort of information big bang. The problem no longer lies in getting the information but more in finding and sorting out the relevant items from similar information. This process is commonly referred to as data mining. Researchers are therefore trying to find various tools and techniques to facilitate the information filtering of the users.

3D-information visualisation is an example of a technique that will add a dimension to allow the users to explore data at a more abstract level that may have no natural visual representation. These systems will let the users take advantage of their cognitive, perceptual, and intuitive skills in order to visualise the data that may be of interest. The many techniques used in information visualisation all use some aspect, property or value of the data items to produce a mapping to objects within the visualisation. See section 5.1

The limited screen display may be used more efficiently which is an important aspect for visualising large amounts of data. Experiments have also indicated that displaying data in 3D instead of 2D can facilitate the understanding of the data for the users [Young 1996].

There are often situations where there is a need to visualise the correlation of two or more variables over time. In these situations, the correlation could be represented as a 3D graph with time projected on the third axis. Another name of such a graph is a 3D surface, which can be used to spot trends in the data. Customers of ABB Network Partner using S.P.I.D.E.R have asked for this form of information visualisation.

8.2.2 Attractiveness

With the current development of hardware for 3D graphics and the development in the computer games market it seems inevitable that companies will also like to incorporate 3D graphics in their business applications. Future applications should have a more modern look so that users will have the feeling that they are using something new and attractive. The users of today do not want their interfaces to look the same as they did in the beginning of the eighties. Nor will the users of the future be content with interfaces that look like the interfaces of today. The wanted situation in this context is to develop applications that embrace this new technology, not to miss out on its new opportunities. A modern look and feel of the interface will no doubt attract users. This can be seen in the field of computer games where 3D modelling has had a tremendous impact on the industry (See section 5.5).

8.3 New User Interface Design

This section discusses how the prototype conforms to the cognitive aspects of user interface design (see chapter 2) and whether the prototype satisfies the wanted situation (see section 8.2).

8.3.1 Navigation

One of the possibilities of 3D interfaces lies in the ability to construct models of real world objects, as described in section 7.1. With the help of modelling software it would e.g. be possible to create a complex model of a building with all different parts included. This model could be used to aid a service technician that is sent to the building. In one possible scenario, the service technician could be working in the field, receiving messages of errors in buildings through his portable computer. The portable computer could be used to display a 3D model of

the building thereby aiding the technician to quickly find his way to the correct part in the building when he arrives. Another scenario includes a broken power line somewhere out in the countryside. A 3D model connected to a geographical database could aid service personnel in knowing what the site will look like when they arrive.

In our prototype, we have taken a general approach in an effort to facilitate navigation in large tree hierarchies. We have chosen not to remove the old tree view but to keep it and combine it with a 3D display (see figure 8-1). The 3D display can be assigned to display any 3D model that has a connection with the objects stored in the tree. In our example the 3D model consists of a map of Sweden with a few fictive power stations. The leaf nodes of the tree consist of arbitrary objects situated inside the stations. This applies to the Parameter Setting Tool in which case these objects would represent terminals e.g. REC 561s (see Appendix A).

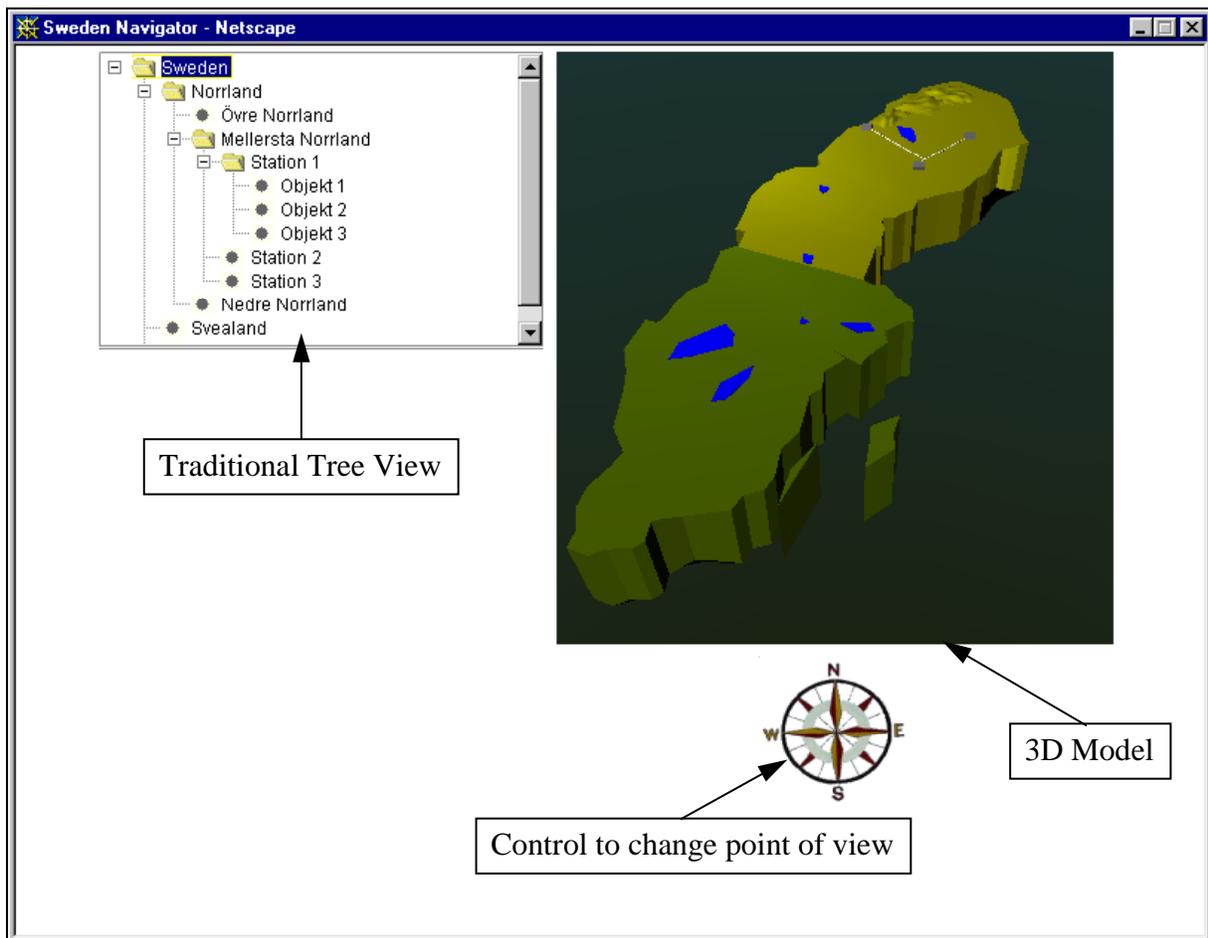


Figure 8-1 User interface of the prototype.

The main goal of the user of this interface is to find a specific node in the tree. This can be accomplished in several different but equivalent ways. The traditional way of navigating the tree i.e. to expand the tree by clicking on the handles of the nodes is still available. When the user clicks in the tree the 3D display changes to reflect the current selection. The display performs a fly-over above the map when it moves from point to point, it animates smoothly so that the user can track the movement and obtain an understanding of how objects are connected.

Another way of navigating the tree is to point and click on areas in the 3D display (see figure 8-2). When the user moves the mouse pointer over the map, the underlying area changes

colour to indicate where the view will move if the user would press the mouse button. When the user navigates the map in this way, the selection in the tree changes accordingly. A third way of using the interface is to move the viewpoint freely around in the 3D model by using the keys on the keyboard. This way of navigating is suitable for exploration and for getting a general view of which objects are connected in the system.

By displaying the map of Sweden in 3D, we have added one dimension to the information visualisation. This has possibly made it easier to survey connections between objects by using artefacts from the real world. The introduction of the 3D-map also support the users' natural ability to create cognitive maps (see section 2.1.1). Natural ways of navigating the map, such as using landmarks, paths, and other generic components are available. As a further aid to the user when navigating the map, all transitions between viewpoints are smoothly animated. We have noticed that this reduces the risk of losing context when moving around. As stated in section 2.1.2, it has been shown that users' spatial ability is relevant to their ability to make use of computer systems. Since spatial abilities are also relevant when using our prototype, this relation will still hold.

We believe that the introduction of 3D graphics and animations has had a positive effect on the attractiveness of the prototype. This attractiveness may fade in the future because of an increasing use of 3D interfaces.

8.3.2 Interaction

To attract the attention of the users different techniques are used. In the case of an alarm going off at a power station, the station changes colour and starts flashing. In future implementations this could be accompanied by auditory signals. Other ways of getting the users attention, described in section 2.2.1, are colour, shape, texture, shading, size, and location, which are all prominent in the 3D display.

Using the principles of affordance (see section 2.2.2), parts of the map change colour when pointed at with the mouse to indicate that they are possible to choose. Also in line with affordances the interface allows clicking on almost all interface component making it possible for different users to use the prototype in the way he or she prefers.

The prototype uses abstraction, as described in section 2.2.3, by providing different levels of detail. At higher level smaller details are invisible and can only be reached by zooming the map. Power stations are e.g. represented as boxes when viewed from a distance, but in a close up view different parts in the power stations are available.

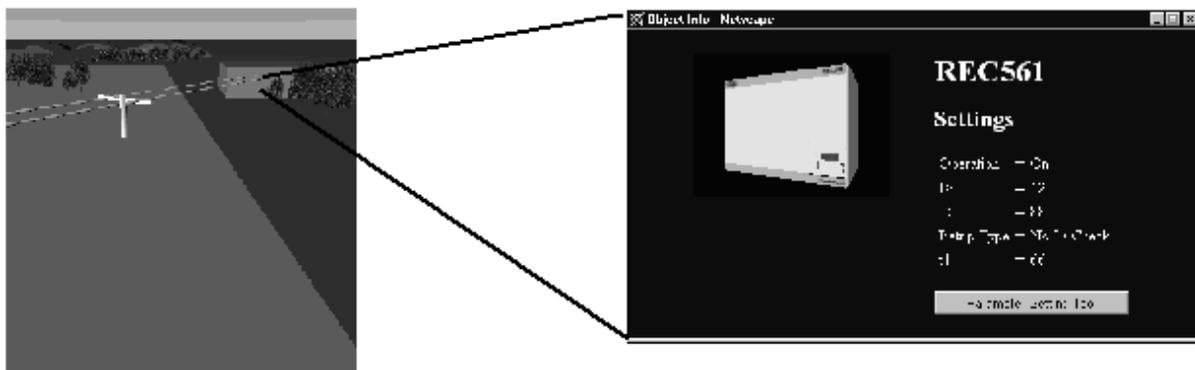


Figure 8-2 Parameter settings become available by clicking on objects in the 3D display.

8.4 System Architecture

To be able to develop a prototype quickly, we chose to work with the programming language Java in combination with VRML (see section 6.2). VRML enabled us to focus on designing the user interface instead of having to care about a lot of details of 3D rendering.⁵ The prototype consists of two Java applets and a VRML browser embedded in a Netscape browser extended with a VRML plug-in⁶ (see figure 8-3). The EAI (see section 6.2.5) enables communication between the applets and the VRML browser.

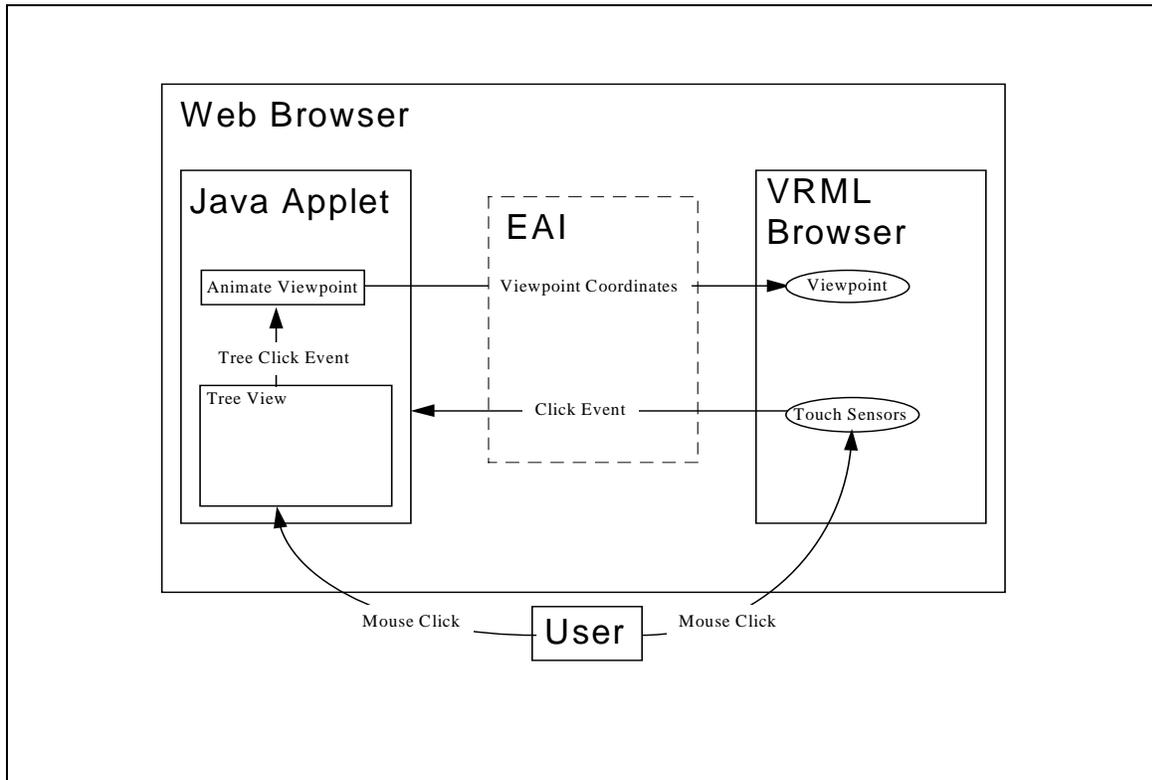


Figure 8-3 General view of the architecture of the prototype.

Loading an HTML file in the Netscape browser starts the prototype. The Netscape browser opens a new independent window containing the Java applets and the VRML browser. Since the prototype is Internet based it enables anyone with a connection to the Internet and a VRML compatible web browser to run it.

⁵ The 3D model was created using the modelling tool Spazz 3D (<http://www.spazz3d.com/>)

⁶ <http://www.cosmosoftware.com/>

9 Summary

The purpose of the thesis was twofold: to conduct a survey of 3D GUIs from a cognitive perspective and to investigate how to implement 3D GUIs.

In our survey of 3D GUIs, we showed that 3D GUIs can provide a more convenient way of navigating through computer systems. This is because of their possibility to take advantage of users' natural ability to create cognitive maps. Still, one must bear in mind that this has not yet been proven. The added third dimension of 3D GUIs may facilitate navigation, but may also simply provide new ways of getting lost.

User interaction with 2D-interface components is based on an understanding of the affordances of the components. No such commonly accepted understanding exists for 3D GUIs. Much research is needed in the area for this problem to be solved. We believe valuable insights in these matters can be obtained by studying modern computer games. Advantages of 3D interaction lies in its potential to take advantage of differences in visual appearance, such as shape, shading, texture etc.

The interfaces nowadays referred to as 3D GUIs are almost exclusively "hybrids" between 2D and 3D interfaces. True 3D interfaces i.e. interfaces with all its components in a 3D environment has not yet had any major impact outside the laboratory. Current application areas for 3D GUIs include information visualisation, simulation, CAD, CSCW, and entertainment. We have further identified three main areas where 3D GUIs can be applied to improve company processes and products. These areas are information visualisation, marketing, and virtual workplaces.

In our investigation of how to implement 3D GUIs we discovered that the use of 3D graphics has been limited not only by the performance of computer hardware but also by the lack of programming standards. We therefore focused on two standards that represented different approaches to standardisation of 3D graphics; VRML and OpenGL. VRML can roughly be described as a 3D counterpart to the World Wide Web standard HTML. It is suitable for creating simple models that are to be published on the Internet. OpenGL is a low-level API suitable for creating more complex applications. Finally we decided that VRML because of its simplicity was the preferable choice to implement our prototype. The main result of this thesis is our prototype. The prototype illustrates how 3D graphics can be used as a complement to 2D GUIs to facilitate navigation in tree structures.

We are of the opinion that 3D graphics will influence the way people interact with computers outside the traditional areas of e.g. CAD and games. There are several possible advantages using 3D graphics, interesting aspects are the possibilities of natural, intuitive navigation, and interaction with computer systems. We are not stating that 3D interfaces are the grand solution to interface problems but it will influence new ways of using computers. We believe the world is not quite ready for true 3D user interfaces, but 3D graphics can very well be used as a complement to 2D interfaces to a greater extent than today.

Bibliography

“A man will turn over half a library to make one book”

Samuel Johnson quoted in Boswell’s Life of Johnson, Chap. viii, 1775.

- [ABB 1998] ABB Network Partner AB, 1998, *User’s manual for CAP 531 VI.3*, 1MRK 511 056-UEN
- [Andrews 1994] Keith Andrews, 1994, *Spatial Metaphors for Information Systems*, ECHT94 Workshop on Spatial Metaphors
- [Boyd and Darken 1996] Casey Boyd and Rudy Darken, *Psychological issues of virtual environment interfaces a CHI 96 Workshop*, SIGCHI Bulletin Vol. 28 nr 4
- [Burton et al. 1997] Nick D. Burton, Alistair C. Kilgour, and Hamish Taylor, 1997, *A case study in the use of VRML2.0 for marketing a product*, Department of Computing & Electrical engineering Heriot-Watt University
- [Carey and Bell 1997] Rick Carey and Gavin Bell, 1997, *The Annotated VRML2.0 Reference Manual*, Addison-Wesley Pub Co
- [Carey et al. 1997] Rikk Carey, George S. Carson, and Richard F. Puk, 1997, *The Development of the VRML 97 International Standard*, http://www.vrml.org/about/vrml_p1b.html
- [Chen and Czerwinski 1998] Chaomei Chen and Mary Czerwinski, 1998, *Spatial Ability and Visual Navigation: An Empirical Study*, Department of Information Systems and Computing, Brunel University, UK
- [Chuah et. al 1995] Mei C. Chuah, Steven F. Roth, Joe Mattis, and John Kolojechick, 1995, *SDM: Selective Dynamic Manipulation of Visualizations*, School of Computer Science Carnegie Mellon University
- [Conner et al. 1992] D. Brookshire Conner, Scott S. Snibbe, Kenneth P. Herndon, Daniel C. Robbins, Robert C. Zeleznik, Andries van Dam, 1992, *Three Dimensional Widgets*, Computer Graphics (Proceedings of the 1992 Symposium on Interactive 3D Graphics), ACM SIGGRAPH
- [Corder 1995] Michael R. Corder, 1995, *NASA AMES RESEARCH CENTER VERTICAL MOTION SIMULATOR*, Sept.issue of PACIFIC FLYER
- [Darken and Sibert 1993] Rudy P. Darken and John L. Sibert, 1993, *A Toolset for Navigation in Virtual Environments*,. Proceedings of ACM User Interface Software & Technology
- [Davis 1991] William S Davis, 1991, *Concepts 3rd ed*, The Benjamin/Cummings Publishing Company, Inc,
- [Embretsen 1998] Joachim Embretsen, 1998, *Development of Relay Setting*

- Tool for Protection Terminals*, Master of Science thesis work for the Engineering Physics program at Uppsala University
- [Gabriel 1985] Pavel Gabriel, 1985, *Oscar Reutersvärd omöjliga figurer i färg*, Bokförlaget Doxa AB
- [Gibson 1984] William Gibson, 1984, *the Matrix*, Ace Books
- [Goldberg and Hiller 1995] Stephen L. Goldberg and Jack H. Hiller, 1995, *Simulator Sickness in Virtual Environments*, Technical Report 1027, U.S. Army Research Institute for the Behavioral and Social Sciences
- [Govil-Pai 1998] Shalini Govil-Pai, 1998, *Learning Computer Graphics: From 3D Models to Animated Movies on Your PC*, Springer Verlag.
- [Hand 1997] Chris Hand, 1997, *A Survey of 3D Interaction Techniques*, The Eurographics Association, Blackwell Publishers
- [Hinckley 1996] Ken Hinckley, 1996, *Haptic Issues for Virtual Manipulation*, In Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy (Computer Science), University of Virginia
- [Höök and Dahlbäck 1997] Kristina Höök and Nils Dahlbäck, 1997, *Designing navigational aid for individuals*, CHI 97 Workshop om Navigation in Electronic Worlds,
- [Jul and Furnas 1997] Susanne Jul and George W. Furnas, 1997, *Navigation in Electronic Worlds: A CHI 97 Workshop*, SIGCHI Bulletin Vol. 29 nr 4
- [Kallinikos 1996] Jannis Kalinikos, 1996, *Organizations in the Age of Information*, Academia Adacta AB
- [Kilgard 1993] Mark J. Kilgard, *OpenGL and X, Part 1: An Introduction*, The X Journal November/December Issue 1993
- [Kotler 1997] Philip Kotler, 1997, *Marketing Management*, Prentice Hall
- [Lampton 1994] Christopher Lampton, 1994, *Gardens of imagination*, Waite Group press
- [Marrin 1997] Chris Marrin, *Proposal for a VRML 2.0 Informative Annex - External Authoring Interface Reference*, <http://www.vrml.org/WorkingGroups/vrml-eai/ExternalInterface.html#1.1>
- [Marrin et al 1997] Chris Marrin, Rick Carey, and Gavin Bell 1997, *ISO/IEC 14772.1:1997 Virtual Reality Modelling Language (VRML97)*, The VRML Consortium Incorporated
- [Media Lab 1998] Media Lab, 1998, *Master of View*, Media Lab University of Art & Design Helsinki UIAH, <http://www.mlab.uiah.fi/nmc/stereo/masters/>

- [Mullet et al 1995] Kevin Mullet, Diane J. Schiano, George Robertson, Joel Tesler, Barbara Tversky, Kevin Mullet, 1995, *3D or Not 3D: "More is Better" or "Less is More"?*, In proceedings of CHI 95
- [Neider et al. 1993] Jackie Neider, Tom Davis, Mason Woo, 1993, *OpenGL Programming Guide*, Addison-Wesley
- [Norman 1988] Donald A. Norman, 1988, *The Design of Everyday Things*, Doubleday
- [Preece et al. 1994] Jenny Preece, Yvonne Rogers, Helen Sharp, David Benyon, Simon Holland, Tom Carey, 1994, *Human-Computer Interaction*, Addison-Wesley Publishing Company
- [Sanders and Gay 1996] Donald Sanders and Eben Gay, 1996, *VRML-Based Public Education—an example and a vision*, <http://www.vrmlsite.com/dec96/spot2.html>
- [Satalich 1995] Glenna A. Satalich, 1995, *Navigation and Wayfinding in Virtual Reality: Finding Proper Tools and Cues to Enhance Navigation Awareness*, Master of Science thesis in Engineering, University of Washington
- [Segal 1994] Mark Segal, Kurt Akeley, 1994, *The Design of the OpenGL Graphics Interface*, Silicon Graphics
- [Simon 1984] David E. Simon, 1984, *GRAFIK till din PC*, Pagina Förlags
- [Smith 1993] Ronald E. Smith, 1993, *PSYCHOLOGY*, West Publishing Company
- [Smith et al. 1982] David Canfield Smith, Charles Irby, Ralph Kimball, and Bill Verplank, 1982, *Designing the Star User Interface*, Byte
- [Stål and Andersson 1997] Olov Ståhl and Magnus Andersson, 1997, *DIVE - A Toolkit for Distributed VR Application*, <http://www.sics.se/dce/dive/online/ercim.html>
- [Wernecke 1994] Josie Wernecke, 1994, *The Inventor Mentor : Programming Object-Oriented 3d Graphics With Open Inventor*, Addison-Wesley
- [Wiss and Carr 1998] Ulrika Wiss and David Carr, 1998, *A Cognitive Classification Framework for 3-Dimensional Information Visualization*, Research report LTU-TR--1998/4--SE, Luleå University of Technology
- [Young 1996] Peter Young, 1996, *Virtual Reality Systems*, Centre of Software Maintenance, University of Durham, <http://www.dur.ac.uk/~dcs3py/pages/work/documents/index.html>

Appendix A

User Interfaces of Applications of ABB Network Partner

To identify potential application areas for 3D interfaces within ABB Network Partner we have studied the user interfaces of several of the companies existing applications.

We will look at four different applications, Configuration and Programming Tool 531, S.P.I.D.E.R., Parameter Setting Tool, and Integrated Data Engineering System/Restructured Data Engineering.

Appendix A gives a brief description of the functionality and user interfaces of the applications

Configuration and Programming Tool 531

The configuration of protection and control terminals in power stations is carried out by a tool called the Configuration and Programming Tool 531 (CAP531), which allows the user to configure the terminal by manipulating graphical symbols.

CAP 531 is used throughout all stages of a project, from engineering to testing, commissioning, documentation, and maintenance. The user can remove and add connections between different function blocks in order to achieve required functionality. A number of free logical elements (AND, OR, Timers, etc.) enables configuration of different customer-specific solutions. [ABB 1998]

Various function blocks can be combined either as pre-determined or custom-designed schemes. This means that an output signal from one function block can be used as an input signal to another function block. External signals can be used to block or enable a certain function.

A monitoring function offers an online check of all internal signals in an object terminal. This function offers the user a powerful help tool by which the user can see the changes in a signal status.

Besides the obvious tool-, status-, and menu bars in The CAP 531 interface, the main screen comprises three different views: the project tree, the work sheet and the page layout.

The project structure, as shown in figure 1, is displayed as a tree, and allows the users to freely build up the structure and select the number of levels. From the project tree menu bar and toolbar, the user can insert items and edit the project tree, copy work sheets and copy terminals.

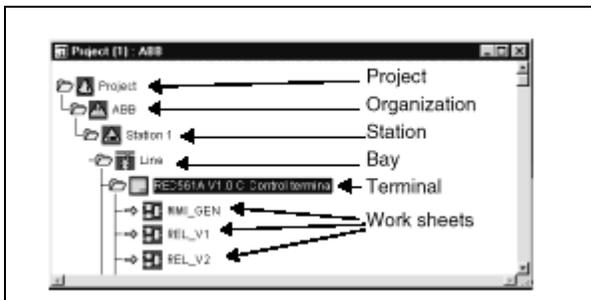


Figure 1 the project structure of CAP 531 is displayed as a tree.

The configuration work is done in work sheets. All functions in the terminal are entered in the work sheet as function blocks. All internal logical interconnections between the functions are drawn as lines between the function blocks (see figure 2). From the work sheet menu bar and toolbar, the user can perform actions to quickly create configurations. The user can also debug the configuration in the work sheet. This is an online function that displays all signal levels of all internal signals in a terminal with different colours of the lines in the work sheet.

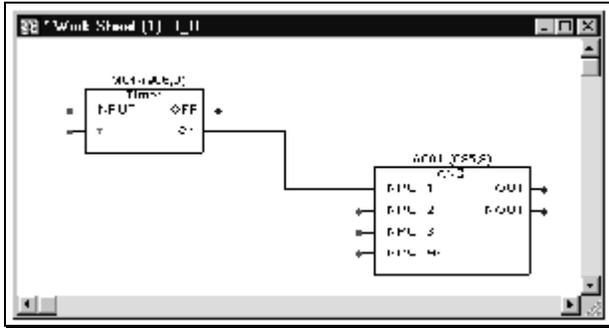


Figure 2 The two objects are connected in the work sheet.

Finally, the page layout is the drawing form of the configuration drawings. It is used to adjust the drawing format of the work sheet pages that are going to be printed. For example, allowing the user to adapt the header and footer and insert order number, terminal type, names, the printing date and time, bitmap files etc.

S.P.I.D.E.R.

The Supervise Power Infrastructure to Develop Efficiency and Reliability (S.P.I.D.E.R.) concept establishes a framework for information management in a power company. It constitutes the kernel for horizontal integration when connecting centralised computing resources. S.P.I.D.E.R. also facilitates the vertical integration by connecting plant-computing resources in power stations and substations to a central position for monitoring and control of all aspects of the electronic network. This spans the entire network stretching from power generation through interconnected transmission grids to distribution networks. [ABB 1998]

S.P.I.D.E.R. consists of a graphical user interface and by using direct manipulation a user can pan the map of an entire network. In an effort to avoid clutter, the user can zoom in or out of the map to change the level of detail and thereby display or hide objects. All functions are controlled using pull-downs and pop-up menus or icons.

Various background maps can be entered as a data file into the system and used as a basis for the network presentation. A network colouring function displays the real time switching state of the network. Changes of a switching device will cause an immediate recolouring on the map. The network can also be coloured according to the types of cables, strong and weak conductors, etc. If a station is chosen, a wiring diagram of the station containing parameters and data is displayed with graphical symbols representing different objects in the station (see figure 3). This allows the operator to monitor and control a wide range of electricity supply functions.

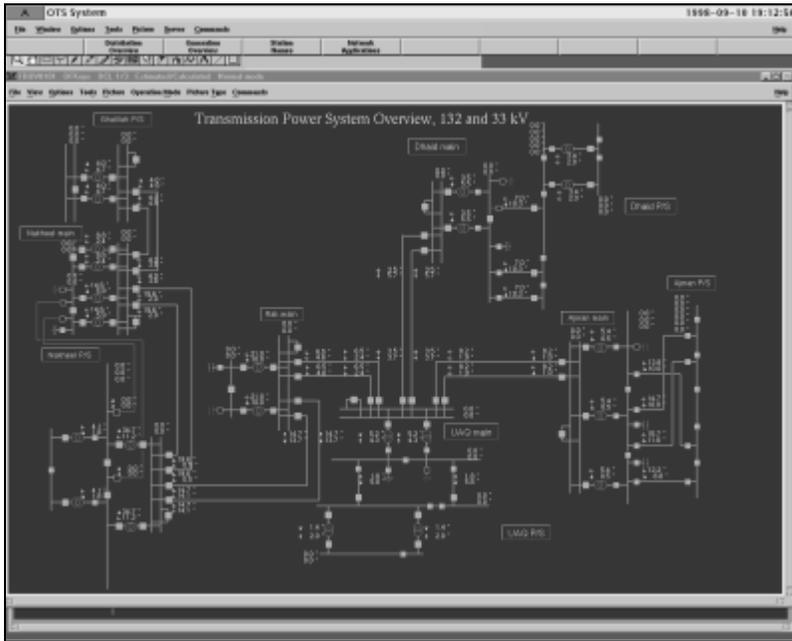


Figure 3 Screenshot from S.P.I.D.E.R.

Parameter Setting Tool

Programmable Logic Controllers (PLCs), are used in power transmission contexts for protection and control of power stations and high-voltage lines. The PLCs are located in power stations and are used to check the currents and voltages in the grid. ABB Network Partner is manufacturing them as protection and control terminals. Substation Monitoring System (SMS) is one of the tools used for handling, storing, and monitoring the information inside the terminals. The information includes amongst other things parameters. A prototype called the Parameter Setting Tool (PST) is a web based improved subset of SMS that is used for reading and writing these parameters. [Embretsen 1998]

A power system can consist of hundreds of stations, each station including up to twenty bays, and thousands of PLCs for control and protection. The PST organises the power system using a tree structure called the Navigator.

Users navigate through the structure by expanding the nodes in the tree. The Navigator in PST reminds of the Windows NT Explorer (see figure 4). Different parts of the structure can be physically located in different places, but this is transparent to the user. One part can for example be located on a central server while another resides on a station computer. From the users point of view it looks as if the whole structure is residing on the same machine.

When the user has navigated through the tree, he finally reaches some kind of leaf that represents some kind of tool functionality. When the user clicks on that leaf the Form Manager starts, which presents the parameters in the terminal corresponding to the node. Parameters are presented on a form and are grouped with tab strips and headers. A description of each parameter is available through a tool tip that appears when the cursor is over a control. Writing in the edit fields modifies the values of the parameters.

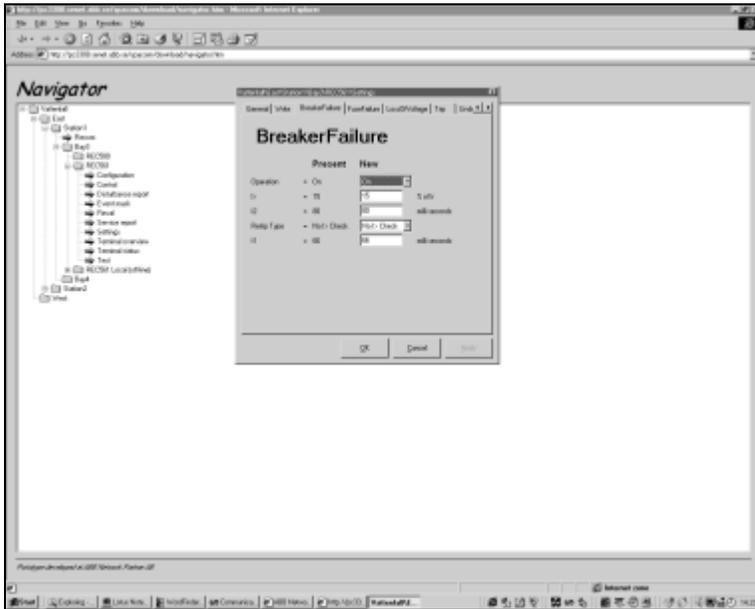


Figure 4 A user navigates through the structure of the Parameter Setting Tool by the nodes in the tree, and the parameters are presented on a form.

Integrated Data Engineering System/Restructured Data Engineering

Integrated Data Engineering System/Restructured Data Engineering (IDES/RDE) is a merger of two data-engineering systems. Both systems are used to design the tables and fields of the database used in S.P.I.D.E.R.⁷

Integrated Data Engineering System (IDES) facilitates database design by providing a graphical user interface, but to insert the database in the S.P.I.D.E.R. system extensive conversion routines have to be used. Restructured Data Engineering (RDE) on the other hand uses a form based input that results in a database that is very similar to the one used in S.P.I.D.E.R.. In an effort to combine the best parts from both systems, IDES/RDE is designed to use the GUI from IDES and the database from RDE. The GUI uses direct manipulation with graphical symbols representing different objects in the database. A completed definition of a database faintly resembles a map with power stations represented as squares with lines between them to show connections. Inside power stations are other objects with differing graphical representations (see figure 5).

To prevent cluttering of the display the map contains multiple levels of detail. Each graphical object has an attribute to indicate on which level of detail the object will be visible. It is up to the user how he wishes to use this feature, but on the top-level usually only power stations and their connections are visible. When the user scales the map to a lower level of detail other objects are revealed e.g. objects inside power stations.

With IDES/RDE it is also possible to edit the database from different control centres simultaneously. To prevent conflicts and loss of data when two users try to edit the same fields in the database simultaneously a system of locked areas is used. To edit a part of the database the user selects a part of the map with the mouse to lock it and make sure no one else

⁷ The information in this section originates from a demonstration of the application carried out by Åsa Mosskull, ABB Network Partner 1998.

can edit that area. To other users, the locked areas are visible as areas that differ in colour from the background. Each graphical object is also colour-coded to indicate from which site the object originates.

To facilitate the input of data regarding power stations the background of the GUI can be set to display a map of the region so that object placement on the screen has a connection to the real world.

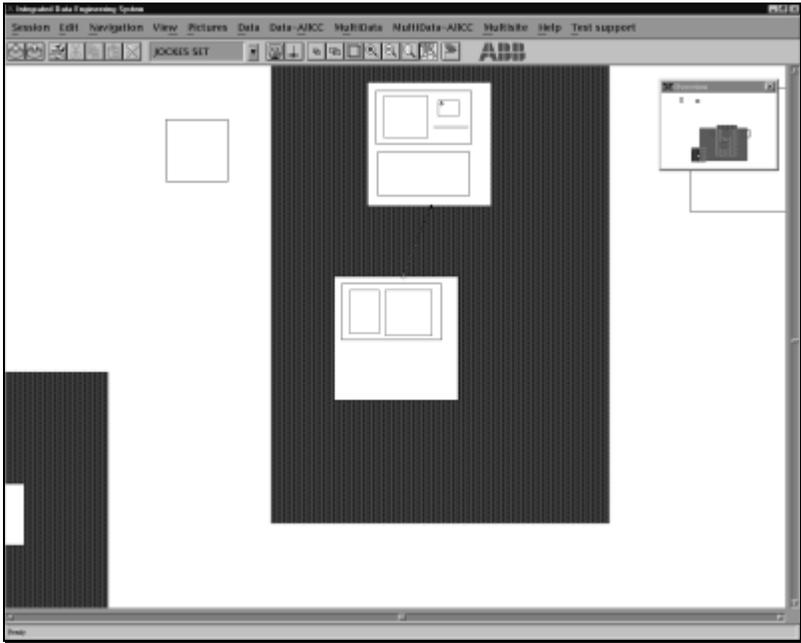


Figure 5 Screenshot from IDES/RDE.

Appendix B

Glossary of Abbreviations

2D	Two Dimensional
3D	Three Dimensional
ABB	Asea Brown Boveri
API	Application Programming Interface
CAD	Computer Aided Design
CAP	Configuration and Programming Tool
CBI	Character Based Interface
CSCW	Computer Supported Cooperative Work
DIVE	Distributed Interactive Virtual Environment
DOF	Degrees Of Freedom
EAI	External Authoring Interface
FSN	File System Navigator
GUI	Graphical User Interface
HCI	Human computer Interaction
HMD	Head Mounted Display
HTML	HyperText Markup Language
IDES	Integrated Data Engineering System
OS	Operating System
Pixels	Picture Elements
PLC	Programmable Logic Controllers
PST	Parameter Setting Tool
RDE	Restructured Data Engineering
SMS	Substation Monitoring System

VE	Virtual Environments
VR	Virtual Reality
VRML	Virtual Reality Modelling Language
WIMP	Windows-Icons-Menus- Pointer
WYSIWYG	What You See Is What You Get