BOOM CHAMELEON: SIMULTANEOUS CAPTURE OF 3D VIEWPOINT, VOICE AND GESTURE ANNOTATIONS ON A SPATIALLY-AWARE DISPLAY

by

Michael C. Tsang

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Abstract

Boom Chameleon: Simultaneous Capture of 3D Viewpoint,
Voice and Gesture Annotations on a Spatially-Aware Display

Michael C. Tsang
Master of Science
Department of Computer Science
University of Toronto
2002

We describe an exploratory design study that integrates a novel input device with the simultaneous capture of annotations in a 3D domain. The goal of this study is to map the design space and identify the areas in need of further investigation. The annotation application we build as a tool for exploration is one of the main results of our work, exploiting the unique configuration of the *Boom Chameleon* device in order to simultaneously capture viewpoint, voice and gesture information. Design issues and results of informal tests on the device and annotation application are presented, and the results show that our annotation facility has the potential to be an effective, easy to use 3D design review system.

Acknowledgements

Many thanks go to the many people who made the research possible: My co-supervisor, George Fitzmaurice, who guided and encouraged me with his patience and video producer/director skills. Bill Buxton, my other co-supervisor, for intuitions that lead to this thesis and his incredibly fast-paced discussions and feedback. My external reader Ravin Balakrishnan, for frank discussions and valuable feedback on research ideas. Azam Khan for advice and help with figures and research. Gord Kurtenbach, for many suggestions and ideas that have helped form this thesis. Scott Guy, for creating the model that was used in the user tests. Angela Morris, for valuable feedback on writing style and semicolon use; and Miles Menegon, for assistance in the creation of the demonstration video.

Alias|Wavefront provided resources for this thesis and their contributions are appreciated. I also thank Fakespace Labs for donating hardware drivers, and the province of Ontario for providing support for this research.

I thank all of the teachers that have taught me over the years. I am especially grateful to the people who introduced and have kept me interested in Human Computer Interaction, especially Kelly Booth, Sid Fels, Oliver Schoenborn, Izak Benbasat, and Ron Baecker. Special thanks go to Joanna McGrenere, Joe Laszlo, Mike McGuffin, Gonzalo Ramos, and other members of the DGP lab for their valued discussion and guidance.

Most of all I thank my parents, who have provided endless encouragement and advice. They have provided the nurturing foundation that has made everything possible. This thesis is dedicated to them.

Table of Contents

List of Figures

List of Appendices

Chapter One:
Introduction

In the realm of digital media, three-dimensional data has come of age. The use of 3D models is prolific, ranging from industrial design to scientific visualization. Tools for the creation and modification of complex models have become incredibly sophisticated and powerful. Yet the technology for annotating these models has been left far behind, resulting in a gaping discontinuity in the high-technology workflow. Due to the widespread use of 3D data, an effective annotation system that addresses this gap is likely to prove extremely useful in industry and offers exciting design opportunities for creating a compelling and interactive user experience.

1.1 Research Background and Motivation

In past work conducted with Alias|Wavefront[1], automotive designers have stated that reviews and critiques of digital 3D models can be awkward and unwieldy. An example is a representative automotive design review where managers and designers are grouped together in front of a large display. The manager critiques the design from various viewpoints but must rely on a technician to operate the interface and move the 3D model. Meanwhile, the designers furiously scribble the manager's comments into a notebook, marking-up images of the 3D model or making quick sketches to capture the manager's intentions. This scenario highlights three problems:

[1] Alias|Wavefront™ is a 3D graphics technology company that develops solutions for the film, video, game development, interactive media, industrial design and visualization markets. Website: http://www.aliaswavefront.com

Viewpoint control – Managers and designers feel frustrated because they cannot directly control the viewpoint. This contrasts with the evaluation of a physical model where anyone is free to move around the model and direct the group's attention.

Information capture - Designers feel that critique information is lost because attention is divided between recording the annotation and participating in the discussion.

Shared view - Both managers and designers feel that misunderstandings could be reduced if notes and annotations could be created and viewed by all parties simultaneously.

Outside the realm of 3D visualization, a substantial body of literature exists on supporting reviews and critiques through digital annotations on text documents, images and other 2D assets. The Wang Freestyle system [50] illustrates an approach of capturing existing work-practice annotation styles, such as marking on documents or simply speaking. The Wang Freestyle system not only captures voice and markings simultaneously, but *synchronizes* them. Although system has not been formally evaluated, we believe that a key to the success of this approach is because the capture of information is minimally intrusive. This is the case because the user does not need to constantly switch modes to record different types of information. In effect, all data streams are constantly recorded, removing the burden from the user of having to explicitly engage different annotation tools. We intend to augment the approach with viewpoint information, and apply it in a 3D domain.

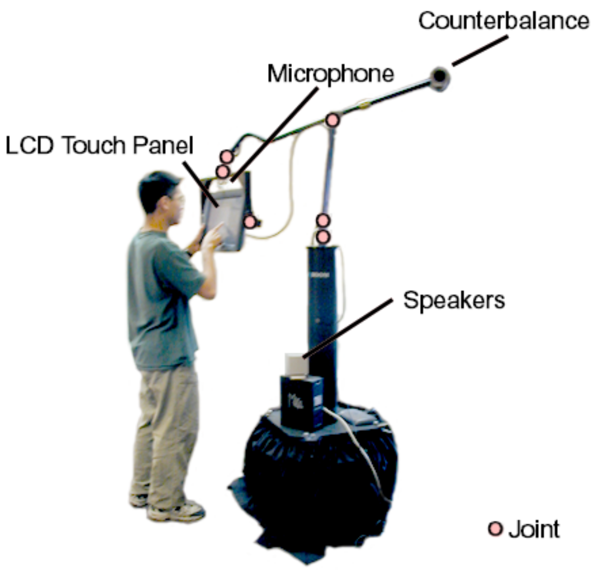


Figure 1 - The Boom Chameleon: a five degree-of-freedom spatially aware display.

Over the past few years the research group at Alias|Wavefront has been experimenting with a specialized device for 3D visualization called the *Boom Chameleon* (see Figure 1). Created in 1999 by George Fitzmaurice, the Boom Chameleon is a 'spatially-aware' device [26, 27], where the display acts as a window onto a virtual world and moving the display in space changes the viewpoint into this virtual world. This idea can also be thought of as a 'video camera' metaphor.

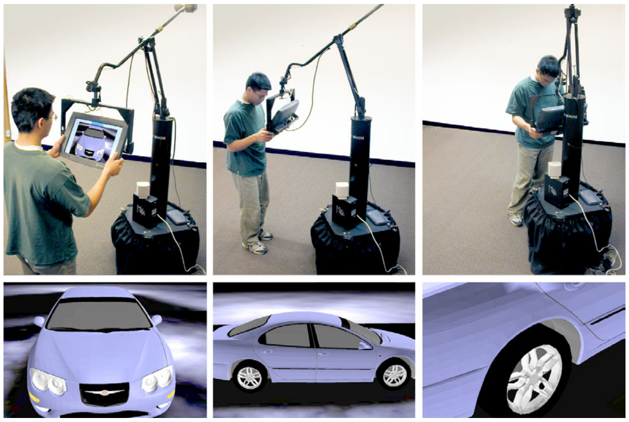


Figure 2: Navigating with the Boom Chameleon. When a user moves to the front of the space the front of the virtual car is viewable, moving to the side shows the corresponding side of the car, and moving closer to the center zooms to a close-up of the car.

The Boom Chameleon was designed to be a strong specific device [13], meaning that it performs extremely well in a narrow application domain, in contrast to a weak general device[2]. Viewpoint controls typically found in commercial applications have discrete functions for zooming, panning, tumbling, etc. In contrast, the Boom Chameleon replaces all of these operations with simple physical manipulation of the display. The virtual model remains fixed in physical space while the user moves the display around in that same physical space. This allows an untrained user to move around a 3D model and inspect it from different viewpoints. As Figure 2 shows, when a user moves to the front of the space the front of the virtual car is viewable, moving to the side shows the corresponding side of the car, and moving closer to the center zooms to a close-up of the car. To return to a particular view, the user simply returns to the same physical point in space. Because it uses simple and familiar mechanisms for navigation, the Boom Chameleon is very well suited for allowing non-technical users to control the viewpoint when examining 3D models.

This thesis explores combining the properties of the Boom Chameleon with previous research on the simultaneous capture of 2D annotations to create a tool for the effective review and critique of 3D digital models. This integration seems to address the three major problems brought forward by managers and designers. First, the ease of use and operation of the Boom Chameleon addresses the *viewpoint control problem*. Second, information capture techniques from 2D annotation schemes can be adapted to address the *information capture problem*. Lastly, the display will address the *shared view problem* by serving as a common viewing and annotation space.

1.2 Research Approach, Objectives, and Overview

Approach. The synchronized, simultaneous capture of voice and marking gesture information introduced by the Wang Freestyle is re-implemented and augmented with

[2] A weak general device is a device that can do many things, but many more specialized tools can do better for each of those things, making the general device weak by comparison. Strong specific devices can only do a few things, but it does them very well, which makes the specific device strong. [13]

viewpoint information. We apply this augmented approach to the 3D domain by using the Boom Chameleon, a strong specific device that excels at certain types of navigation (§2.1.2).

With this novel combination of annotation philosophy and hardware, we engage in an exploratory survey to carve out a rich, multi-dimensional design space. We do not focus on a particular area of this space by performing formal user evaluations. Instead, we attempt to shed the most light on the topic with the resources available, so we can identify the areas of interest that warrant more rigor and formal evaluation.

Objectives. The exploratory research was guided by three main objectives: (1) examine the characteristics of the Boom Chameleon device that make it conducive to certain types of navigation; (2) build an annotation application which makes use of the Boom Chameleon to allow the simultaneous capture of different input modalities; (3) evaluate the users' tendency to make use of the simultaneous data capture during annotation.

All three objectives were met during the course of this research. The application interface was improved through iterative design, and novel techniques were introduced. Informal user testing found that the users successfully integrated the different modes of input during annotation.

Overview. The second chapter of this thesis reviews related areas of research and presents an analysis of the Boom Chameleon. Chapter three presents the interface, features of the system, implementation details, design rationale and insights. In the fourth chapter, we discuss the procedure and results of several informal user evaluations. We also discuss lessons learned, design insights and the general feasibility of the approach. The fifth and final chapter consists of chapter summaries, future directions of inquiry, and the conclusions

Chapter Two:

Background and Related work

The two main areas related to our work with the Boom Chameleon are navigation and annotation. This chapter reviews the background literature in these fields with a focus on related systems and research. The first section reviews navigation and the systems most strongly related to the Boom Chameleon. The second section includes the review of systems relevant to multimodal 3D annotation.

2.1 Navigation

Navigation in three-dimensional spaces is a rich and complex topic. Some studies focus on metaphors used in navigation [37, 81], and the affordances$_{(G)}$[3] of input devices that support those metaphors [84]. Other relevant work examines the effect that constraints may have on navigation, and the resulting biases that may occur. The Boom Chameleon is a physically constrained device that employs 'chameleon-style' navigation. We

[3] The terms 'afford' and affordances' are used throughout this essay and have two meanings in popular use within the field of Human Computer Interaction. One meaning of the term affordance was introduced in 1979 by J.J. Gibson, "…the affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill." [34, p.127] This was described as "an action possibility" by McGrenere [57]. D.A. Norman appropriates the word for a different meaning in 1988, "…the term affordance refers to the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used. A chair affords ('is for') support and, therefore, affords sitting. A chair can also be carried." [62, p.9] This is summarized as "a perceived suggestion" by McGrenere. Both definitions are elucidated upon and, in terms of Norman's definition, disambiguated in [57]. We denote terms in this essay adhering to Gibson's definitions by the subtext '(G)'. Similarly, the use of the term 'afford' or 'affordance' adhering to Norman's usage is henceforth denoted by the subtext '(N)'.

examine the suitability of devices for navigation, the use of constraints, chameleon navigation, and then perform a detailed analysis on the properties of the Boom Chameleon and the biases exerted on the user during navigation.

2.1.1 Input and Output Devices

Although any[4] input device may be used for navigation, the choice of input device is an important factor for the resulting interaction. Chris Hand agrees: he presents a survey of different 3D interaction techniques [37] and shows that the type of input device may strongly influence the techniques, which in turn may affect the navigation. This section discusses the suitability of devices for certain types of navigation.

Aside from the standard keyboard and mouse, some of the most popular input devices for navigation of 3D environments include magnetic trackers [5] and the devices such as the Spacemouse [74]. Hinckley [41] touches upon the importance of device ergonomics among other spatial input design issues.

As with input devices, the choice of visual output devices such as a head-mounted display (HMD) or CAVE™ may have a significant impact on the user interaction, as illustrated by Buxton [13]. However, Bowman [7] claims that there is little empirical data about the relative interaction performances for virtual environment display devices, but there are some general guidelines: CAVEs and HMDs are more suited to immersion, encouraging an egocentric inside-out point of view such as those used in first-person games; workbench displays support more of an outside-in exocentric point of view, which work well for third person tasks like manipulation of an object.

2.1.1.1 Degrees of freedom

Devices with very low degrees-of-freedom are sometimes considered less useful for navigation in higher dimensions, because of the limited ability to convey simultaneous information. In contrast, devices with six degrees of freedom are considered well suited

[4] For example, [43] shows navigation based solely on sound

for navigation in 3D, since they allow the simultaneous specification of multiple values such as x, y, z, pitch, yaw and roll. Zhai investigates the intricacies of such high degree-of-freedom devices in his thesis, including the affordances of different device classes, and the physical properties that allow the task to be learned quicker [84].

On the other hand, many simple tasks may become more complex if an improper choice of input device is made [6]. For example, it is disadvantageous to use a tracked hand to reach a virtual on/off switch when a pinch-glove or a simple physical button enables the same functionality. Following this line of reasoning, a good heuristic is to reduce the unnecessary degrees of freedom whenever it is feasible. [42]

Because the physical characteristics and affordances$_{(G)}$ of the input device are important to the interaction, devices are altered in order to obtain specific properties. For example, because Galyean and Hughes [32] found that the tracker was difficult to control, they attached elastics to the tracker and placed it into a frame.

2.1.1.2 Use of Constraints

Given an input device with extraneous degrees of freedom, one way to reduce complexity of a task is to use constraints. Common approaches are to reduce extraneous degrees of freedom, or to perhaps cull the navigation space by restricting the navigation to a certain area. For example, if a user is only interested in simple translation, it does not make sense to also allow simultaneous rotation using related degrees of freedom, since allowing the rotation would make the user's task more difficult by requiring translation with additional overhead to ensure that no rotation occurs.

Constraints may guide actions and often offload cognitive overhead from the user onto the system. This may be beneficial, as long as the mechanism of applying constraints is not awkward and the constraints themselves are both consistent with expectation and compatible with the user's intentions. Indeed, there is evidence that complex tasks can negatively impact learning, usability and performance [56].

In contrast, if constraints are awkward to employ, the user will have a steep learning curve. Constraints can also lower performance if they do not fit the structure of the task at hand [45]. Furthermore, one could imagine that even if a user does become proficient with a feature, if the mechanisms for applying them are tedious and unwieldy, the frequency of use and subsequent adoption of the system may suffer. Hinckley [41] agrees. He acknowledges that although software constraints are often useful, one limitation is that the user must understand the scope of the constraints and be able to react accordingly to achieve desired results. However, how to go about actually implementing effective constraints is not always apparent.

Rule-based constraints. Constraining virtual movement using rules is a 'soft' method of focusing the navigation. A common example is an exocentric view that constrains the flying metaphor to orbiting around a particular object of interest. Another example of movement constraint would be to disallow the viewpoint from passing underneath the ground plane. There is even an example where a software-implemented bias is used dynamically to compensate for a small physical environment [67].

One example of a system actively integrating navigation metaphor with constraints is described by Santos et al [69]. They describe task-dependent or "metaphor-aware" navigation, where the navigation mechanism is bound to the visual metaphor. The system will limit the user to navigation through the world with the method most suited to the particular metaphor being used. They use the scenario of a path-based navigation in an office building, where the system routes the user through hallways and down stairs rather than instantaneously jumping to a destination. This preserves the user orientation in conjunction with expectation. This case is an example of a system using rules to constrain the available mechanisms of navigation in order to allow for a more recognizable experience for the user.

Physical Constraints. Another method of guiding interaction is by using a 'hard' physical constraint. A physical constraint is distinct from the standard operational range of the device, since a physical constraint is something that is not necessary to the operation of the device, but rather an addition such as a prop to somehow alter the operation of the device. Using physical props "to constrain and disambiguate complex spatial tasks" [7] may be an easy concept for the user to grasp, especially since concrete physical 'real-world' constraints are often more readily understood than invisible software-implemented rules.

One example of using a physical constraint is a physical table to provide a 2D surface for a tracked pen [52, 66]. A tracker embedded in a doll's head that assists surgeons in determining orientation [41] is an example that uses physical props to help with intuition. Physical constraints can also be used to signify locations, guide the space of interaction, and even create simple haptic feedback. Examples include using physical trays and an inkwell metaphor to constrain the mode changes [26], Galyean & Hughes' "poor man's force feedback Polhemus" [32], Zhai's similar "Egg" [84], and Brown, Buxton, Murtagh's stencils on a touch tablet [8]. A physical constraint that is simple to understand and manipulate may create a navigational constraint which is correspondingly transparent and easy to use.

As illustrated above, different aspects of a system will bias the navigation in some way. A device brings unique affordances$_{(N)}$ and characteristics to the interaction, while constraints may bias the navigation by preventing one set of actions and promoting another.

2.1.2 Chameleon Navigation and Related Devices

Chameleon-type navigation is synonymous with the "window in hand" metaphor introduced by Fitzmaurice [26, 27]. Using this metaphor, the device acts as a physical window into a virtual world. A number of systems, including the Boom Chameleon, use this style of navigation. One system closely related to the Boom Chameleon, using both "window in hand" navigation and a supporting armature is the room-scale Art+Com [1] virtual car installation (see Figure 3). A small handheld display is used without an armature by Small and Ishii to navigate virtual newspaper media and other 2D information [73]. The metaphor has also been used with video cameras on handheld devices to view the physical world with computer graphics superimposed on the display [54, 68].

Other work [14] discusses the relationship between the eye, hand, and display for different devices and the impact on the interactions that they afford$_{(G)}$. This work highlights the unique properties of chameleon navigation in relation to other virtual reality approaches such as head-mounted-displays, CAVEs, and the standard workstation configuration.

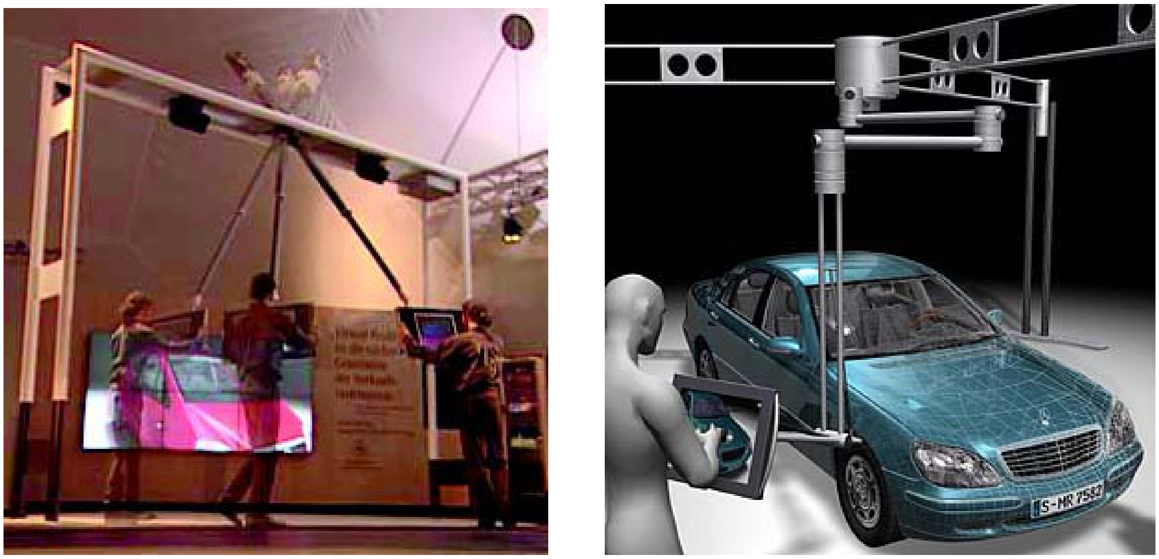


Figure 3: Art + Com. Older and newer generations both use chameleon-type navigation.

The WindowsVR device [83] is a cousin of the Boom Chameleon consisting of a touch sensitive LCD panel, which is hung from an overhead arm using a retractable cable (see Figure 4). The device comes in two configurations. In the first, only the orientation of the display is sensed; to specify position, the user must operate two joysticks mounted on the sides of the LCD. In contrast to the Boom Chameleon, this method of interaction breaks the one-to-one spatial mapping between the physical and virtual space, as the user must navigate by using a flying metaphor. In the second configuration both the orientation and translation are sensed. However, the display can be cumbersome to operate since it is only counterbalanced vertically.

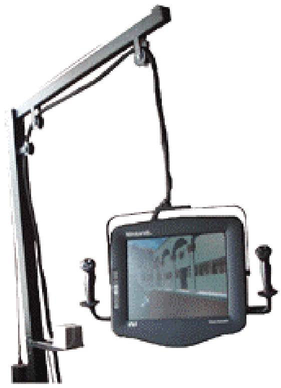


Figure 4: The WindowsVR device is similar to the Boom Chameleon in size.

2.1.2.1 Ease of Use

We claim that using the Boom Chameleon to navigate a 3D space and examine models is easy and intuitive. Although this statement may seem difficult to qualify, previous research supports this assertion. In [11], Buxton suggests that some aspects of “ease of use” such as lowered error rates, reduced frustration and shortened learning time can be better captured by asking “How can we accelerate the process whereby novices begin to perform like experts?”

Indeed, from our informal testing with the Boom Chameleon, we noticed that novice users immediately performed navigation with the same apparent degree of skill as experienced users. Whether the learning process is very fast, or the device is trivial to use, an examination into why may yield valuable design insights:

Buxton [11] argues that the granularity of the task is a major difference between the novice and the expert. Whereas novices tend to spend a lot of time concentrating on lower-level tasks, experts will be more attentive to higher-level ones, while performing the lower level tasks automatically. In order to allow people to automatically perform lower level tasks, he suggests that “there should be as close a match as possible between the structure of how we think about problems and the language or representation that we use in solving them.” In the world of devices, this suggests that the problem domain and the user’s operational perception of the device should be congruent. Buxton concludes: “…interface pragmatics can be designed to accelerate the acquisition of expert operational skills. The key is (physical) gesture-based phrasing to chunk the dialogue into units meaningful to the application. Any concept of transaction that can be described in a single word or phrase should be able to be articulated by a single gesture."

Buxton echoes this sentiment in [11] when he asserts, "…we must learn to match human physiology, skills, and expectations with our system's physical ergonomics, control structures, and functional organization." The main thesis of his paper is of the same thrust: "If you can say it in words in a single phrase, you should be able to express it to the computer in a single gesture. This binding of concepts and gestures thereby becomes the means of articulating the unit tasks of an application".

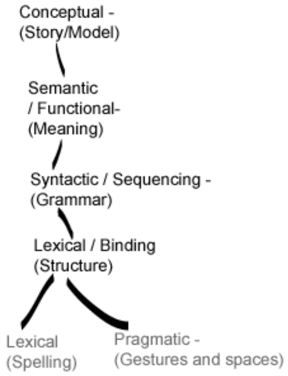

Conceptual - (Story/Model)
Semantic / Functional- (Meaning)
Syntactic / Sequencing - (Grammar)
Lexical / Binding (Structure)
Lexical (Spelling)
Pragmatic - (Gestures and spaces)

Figure 5: A framework that can be used to analyze user interfaces. *Conceptual design* is characterized as the main concept of the system as seen by the user, or the "user's model of the application". *Semantic design* is what can be expressed by the functionality of the system by defining the meanings, but not the actual actions or the manner in which the actions are conducted. *Syntactic design* is the grammatical structure of the action vocabulary used to articulate the semantic concept. *Lexical design* defines the structure of the vocabulary, the binding that determines the meaning of each interaction primitive.

A framework may assist in the analysis as to how the Boom Chameleon allows novices to perform ostensibly complex 3D navigation easily, perhaps as an atomic task. In [30], Foley et al. describe a four-layer structure, first proposed in [29], for the analysis of interfaces. The four tiers are the conceptual, semantic (functional), syntactic (sequencing) and lexical (binding).

Buxton [8] further divides the lexical design into the *lexical* and *pragmatic* components. The *lexical component* is the spelling of the vocabulary or nature of the alphabet; and the *pragmatic component* includes “issues of gesture, space and devices” (see Figure 5). The author’s main thesis is that “since the primary level of contact with an interactive system is at the level of pragmatics, this level has one of the strongest effects on the user’s perception of the system.”

At the risk of straining the metaphor, if this hierarchy related to a storytelling book the following would apply: The conceptual layer is the overall story being told; the semantics are the meaning of each sentence; the syntax is the grammar of words that forms a sentence; the lexical component is the spelling of the individual words; and the pragmatics relate to the size, color, and layout of the letters on the page.

These design strata may help us to understand how the Boom Chameleon articulates the vocabulary of interaction and allows the user to easily express the conceptual task of navigation. At a *conceptual* level, the device allows users to navigate space and examine a model. *Semantically*, the viewpoint is moved around to achieve certain views that help in our examination. *Syntactically*, the display traverses a series of paths and positions to achieve the views. Each path or position is composed of the *lexical* rotations and displacement components that are necessary to achieve the syntactic grammar. The *pragmatic* component includes the manipulation of the device by pushing, pulling and pivoting the device in physical space using the hands.

With this framework as a basis for analysis, we can suggest why it is easy for people to use the Boom Chameleon. Infants invariably become adept at the *lexicon* of grasping food and other objects and pulling it to a specific location (their mouths). We may have learned the *syntax* and *pragmatics* of push, pull and pivoting blocks into holes as children. We understand *semantically* at a very early age that moving the viewpoint in the physical world will change or view, as can be readily observed when a child peers behind a person’s back to see what has been hidden from them. Most of us have had daily practice of planning paths through the physical world in a complex 3D environment. As a result of early exposure and constant practice, the grammar of moving physical objects and changing the viewpoint is automatic for most people. We no longer think about the complex series of muscle contractions needed to move a coffee cup to a different location, nor do we have to ponder about how to move so we can see a squirrel hiding on the opposite side of a tree[5]. We simply perform the action and take the pragmatic, lexical, and syntactic levels for granted. The only levels left to cope with are the conceptual high-level task the examination itself, and the semantic[6] determination of what the different views mean.

As a result of this analysis we can reasonably assert that the Boom Chameleon[7] makes it easy to navigate and provides a natural method for examining models in 3D space.

2.1.2.2 Navigational Bias and Limitations of the Boom Chameleon

The Boom Chameleon not only facilitates easy navigation, but it also biases the navigation in four major ways:

1. The use of a spatially-aware chameleon concept biases the interaction between the hand, eye and display [13]. One difference between the Boom Chameleon and previous chameleon pedigree is the addition of a touch sensitive overlay onto the display [16]. The addition of this touch input takes advantage of the strongly linked relationships between display and hand to engender a low-overhead mechanism to interact with objects in the scene.

2. Although it is based on the free-floating chameleon navigation concept [26, 27], the Boom Chameleon has a display that is constrained by a mechanical armature to a

[5] Most squirrels also understand the function between movement and viewpoint and they will move to thwart you. They will likely do so without pausing to think about the lexical, pragmatic, syntactic, or even the semantic levels of their action.

[6] We will sometimes even forget the semantics, as demonstrated by people who inadvertently walk into walls.

[7] And other devices based on chameleon-style navigation.

person-sized operational range. The armature physically constrains the interaction space inside a spheroid centred on a non-negotiable pole. This has implications for positioning a model and has an observed impact on the contours of user navigation. For example, users tend to go around in arced paths instead of straight lines, and they do not tend to make many sharp-angled turns. Furthermore, looking inwardly at an object centered at the base of the armature prevents the user from interfering and colliding with the arm and, with an object positioned at the center, the base acts as a physical barrier and does not allow the viewpoint to be moved through the center of the object.

The Boom Chameleon differs from other larger or smaller armature-constrained systems because the scale of the device allows for very different affordances$_{(N)}$ for navigating around the space. For instance, a hand-sized device [48] would not have the same sense of immersiveness as a larger-scale device would. On the other hand, a room-sized device such as Art+Com [1] lacks effective focusing properties because the physical constraints of the boom are less apparent than that of the person-sized Boom Chameleon.

3. The Boom Chameleon is a weighted and counter-balanced device. This allows for a different set of positioning affordances$_{(G)}$ than the original Chameleon device, even allowing the user to position the device using only a single finger. The weight and inertia of the device naturally stabilizes the motion and results in a smooth trajectory of motion, dampens small involuntary movements and provides kinaesthetic feedback to the user.

4. Finally, the superposition of a virtual model onto a physical space allows the user to anchor the interaction with the viewpoint in the physical world. This makes use of the user's spatial ability and recollection, enabling virtual navigation using physical affordances$_{(G)}$. This allows the user to establish semi-closed loop navigation with respect to the virtual viewpoint.

As a result of these navigational, view, and interaction biases the Boom Chameleon is conducive to egocentric navigation for object inspection, and is well equipped for relatively simple viewpoint manipulation and object inspection in 3D space. Yet it is interesting to note that many of the biases can be considered both strengths and weaknesses. For example:

- Although the pole acts to focus the interaction around a central location, it also inhibits the user from going through it, causing difficulties if it is the intention of the user to position the viewpoint at the center of viewing volume. Furthermore, while it is also easy to view inwards toward the pole, viewing outwards away from the pole is more difficult because the armature may be an obstacle, and there is only a small space to walk between the base and the display. Solutions to this problem may entail use of a larger armature, inversion of the device by mounting it to the ceiling, or offsetting the origin of the virtual scene.

- The device is inexpensive compared to some virtual reality systems such as a CAVE™, yet expensive compared to the standard 3D workstation.

- When using a simple one-to-one mapping, the user must be able to physically move the display. This means that the user is constrained not only by the device, but by the laws of physics on his or her own body. This leads to performance that is consistent with expectation, but it is unfortunate if the user wants to move to a position that is unreachable, without the ability to fly.

2.2 Annotation

The second half of this chapter discusses the research and systems that relate to annotation. Several systems capture information for annotation purposes. Often-captured modes of communication include speech, and both marking and non-marking gesture[8].

[8] *Note about marking gesture nomenclature:* For the purposes of this essay, the terms ‘marks’, ‘ink mark-up’ and ‘marking gesture’ are interchangeable. All describe the creation of artefacts by the user to convey information, such as when writing with an ink pen. This contrasts with the ephemeral ‘non-marking gesture’.

Specific complexities that arise from annotating in three dimensions are discussed in this section, as well as the advantages of combining different modes of annotation.

2.2.1 Annotation in 3D

The nature of two-dimensional annotations tends to be different than that of three dimensions. The extra dimension results in a much greater emphasis on spatial relationships, and the resulting difficulty in providing an effective method of navigation is an example of one of several complexities that act as a barrier to effective annotation.

The Importance of Navigation during Annotation. A requisite chore with annotations involving visual references is to position the view so that the subject of interest is visible. This becomes especially important if the information of interest is not adequately covered by any particular field of view. Without the appropriate visual framing to reference, the integrity of the voice and gesture annotations is compromised. Thus, navigational support during visually dependent annotation is of great importance because it acts as a cornerstone for the other annotation types.

Viewpoint manipulation in 2D annotation systems typically involves x-y translation controlling two degrees of freedom. In contrast, most 3D systems have six degrees-of-freedom, allowing the user to specify the x, y, z position and orientation using yaw, pitch and roll values. Controlling all these values simultaneously in a straightforward and natural manner is a significant challenge. Furthermore, many 3D views are from a perspective-view as opposed to the typically orthographic-view of a 2D system. The orthographic view presents no perspective differences when viewed from different positions. Although misalignments and occlusion are prevented in orthographic 2D, in 3D the user may encounter these problems in the common perspective view when annotations are viewed from different perspectives. A 3D annotation system would reap huge benefits by employing a viewpoint manipulation mechanism that enables the many complexities associated with a 3D viewpoint to be managed through a cognitively simple, yet effective viewpoint manipulation mechanism.

Some examples of annotation in virtual 3D spaces include the Virtual Annotation System [38], Virtual Notepad [66], annotation in 3D fluid [53], SpacePen, and Redliner [45].

Many annotation systems also embed links to media or documents into 3D space, or bookmark discrete viewing positions as part of capturing and reviewing annotation; these systems are interesting, but not in the scope of research described in this thesis.

2.2.2 Modes of Annotation

There are many modes of communication used in annotation systems. Voice, for example is considered to be a good medium for spontaneous, frank communication. Chalfonte et al. [17] suggests that verbal discourse is often superior to the written, in terms of spontaneous self-correction, justification and explanation. The author also suggests that people are more likely to view the process of speaking as supporting collaboration and the use of non-verbal cues, such as inflection and pitch, as contributing to expressive richness. The authors give an example of an extended sigh followed by a pause; this may convey a sense of fatigue, which is an emotion that would be more difficult to capture in a different medium. Although speech is historically viewed as self-sufficient [64], other modes can also convey information that is beyond what is present in speech. For example, spatial information can be specified by pen input, and the manner of action can be conveyed by a gesture.

There are several systems that record both audio and written marks [79, 80]. In these systems the marks are not only used to index the voice, but they can also be referenced by the voice, and vice versa. Recording both voice and written markings allow these systems to integrate the strengths of both written and spoken communication.

According to Ware [81], the gesture used to link verbal and visual references is called a deictic gesture, or deixis. Deixis is a well-studied form of natural dialogue that allows the user to efficiently express ideas by linking verbal and visual references to convey a single message. This combination of modalities is considered to be the most elementary of linguistic acts, but it has been found to be especially prominent when dealing with spatial data, such as with 3D object inspection [64]. In conversation, finger pointing, glancing, or perhaps a nod of the head often indicates deixis. For example, when presenting information about a formation of flying geese, regions of space or groups of objects may be referred to with a circling motion. A classic illustration of deixis is Bolt's "Put-That-

There" System [5], where a user could tell the system to "move that blue square… over there". Many systems can record speech and markings on top of existing content, allowing for annotations that incorporate deixis.

Multimodal Interaction in a Spatial Domain. Combining different modes of communication such as voice and gesture is termed multi-modal communication. The premise for combining modalities is that one mode may compensate for the weaknesses of another and synergies may arise. Sharon Oviatt, author of the article "Ten myths of multimodal interaction" [64] agrees that there are advantages to using multi-modal methods of communication. According to the article, almost all users prefer to interact multi-modally when they are free to use either speech or pen input in a spatial domain; of these interactions "speak and point" comprises 14% of all spontaneous multi-modal utterances, while pen input is used often to create marks, symbols, and gestures. The author also suggests that pointing deictic gestures account for less than 20% of all gestures. One interesting observation mentioned in the article is that multimodal input does not necessarily take place co-temporarily, but may occur by switching between several modes in rapid succession: "Synchrony does not imply simultaneity" [64].

Oviatt notes that the language used with pen and voice combinations are often more brief, syntactically simple and fluent than for speech alone. As well, clarification may occur by using gesture. Furthermore, Oviatt asserts that speech and pen input or speech and gesture seldom yield redundant information and a well-designed multimodal architecture may elicit the "mutual-disambiguation" of input signals. More importantly, "task-critical errors and disfluent language" have been demonstrated to drop by 36-50% during multimodal interaction [42]. The author asserts that a "strong and nearly universal preference" for interacting multimodally suggests a non-trivial advantage to doing so.

Deixis and gesture annotations often heavily depend on the viewpoint in order to reference the subject. Thus the navigation of models and the ability to achieve the desired viewpoint is paramount to successful communication in the 3D domain. In many annotation applications, the overhead for navigating in 3D is so high that the user's ability to annotate "on top of" the viewpoint may be compromised. In contrast, if the navigation is simple to use, then creating voice and gesture annotations in addition to navigation should not compound difficulties. Rather, the system may become more usable by becoming less intrusive, and removing the mode switching by continuously recording further reduces the cognitive overhead in making annotations. The feature set incorporated into the Boom Chameleon annotation system can hopefully harness the navigation characteristics of the device and demonstrate the advantages of non-moded multimodal communication.

2.2.3 Related Systems

There are several annotation systems relevant to annotation on the Boom Chameleon. This section briefly summarizes the devices and approaches, and highlights the differences between these systems and the Boom Chameleon annotation system.

2D Annotation. There is a wide range of research concerning annotation of 2D media such as images, text, and video. The most relevant work from this area involves modeless free-form hand drawn mark-up and voice annotations. The Wang FreeStyle system [50] is the first system to use voice and gesture input to capture synchronized voice and ink mark-up for documents in an annotation context. Other systems have followed similar approaches for capturing meetings [59, 65]; collaborative writing [38]; video annotation [51, 77]; note taking [79, 80], Post-It™ style notes in virtual space [36, 47] and voicemail [31, 78]. Studies of annotation in traditional media such as paper documents suggest that a free-form strategy is attractive since people show great diversity in how they annotate [55, 63]. Furthermore, there has been evidence that spoken annotations can be richer than textual annotations [17], and that users almost universally prefer to interact multimodally, especially when working in the spatial domain [64].

3D Annotation. The notion of annotating 3D scenes has been explored in many different forms. In terms of commercial systems, these are generally designed for traditional monitor, mouse and keyboard configurations and typically use a fairly structured and modal approach to the interface. For example, the Silicon Graphics IRIS Annotator™ [44] allows annotation of 3D models with several media types from fixed viewpoints, employing explicit modalities for input including keyboard text, audio and video. Other research systems have explored similar methods for embedding structured data into the 3D scene. Loughlin and Hughes [53] use virtual “Post-it” text notes and some systems use structured graphics such as pointer arrows or icons to indicate annotations in the 3D scene [38, 49]. Researchers have observed that these structured inputs are not always sufficient [45] and the ability to free-form draw would be beneficial. Early explorations have been conducted into the ability to draw free-form annotations directly on 3D models [18, 45] and onto 2D drawing planes in a 3D scene [6, 49, 74]. Our work continues to develop along this vein of research.

The Virtual Notepad [66] system uses a head-mounted display and a spatially tracked tablet to control the location of a virtual writing surface in an immersive world. Annotations may be placed in the virtual world by positioning the tablet and writing on its surface. The Boom Chameleon differs from this work in that the display is not head mounted and, consequently, the combination of display/touchscreen allows for a different set of user affordances$_{(G)}$ [13].

The main contribution of the research presented in this thesis over prior work is the combination of a spatially-aware display with the ability to capture voice, gesture and viewpoint information in a continuous, free-form fashion.

Chapter Three:

Design and System Implementation

This chapter describes the design and implementation of the Boom Chameleon annotation application. The design philosophy behind the system is to allow annotations that can be created with minimal overhead and user pre-planning. In order to accomplish this all streams of annotation data are constantly recorded. The capture of annotation data without requiring explicit action for each data type may allow for a richer and more fluid set of annotations, when compared with a highly structured and modal approach to annotation entry. The first section describes the interface, features and design rationale. The second section describes design issues, and the third section describes implementation details.

3.1 Annotation Interface

An annotation application was developed to facilitate design reviews and critiques by taking advantage of the simple navigation metaphor (§2.1.2.1) provided by the Boom Chameleon. The application is meant to enable the easy capture of multimodal annotations.

The overall interaction is modeless, meaning that we continuously record all the three major input modalities of gesture, viewpoint and voice and never switch between them. Although we provide different tools for gesture capture, we do not record one type of information to the exclusion of the others. All three major modalities are always on and streaming.

The graphical interface was designed to allow for a wide range of grip and hand positions (see Figure 6). The wide button record and play strips running along the left and right sides of the display are simple and accessible to using the thumbs from a wide variety of grips. There are also large push buttons along the top of the screen for tools for gesture annotation (Figure 6a, b, c).

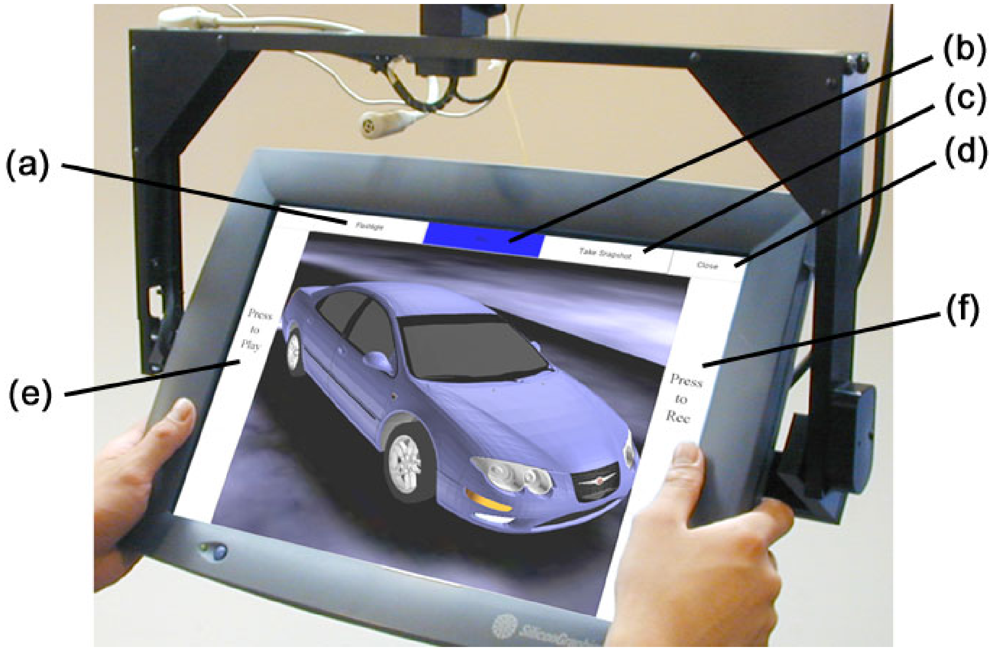


Figure 6: Annotation application hand grip and button layout. (a) Flashlight (b) Pen (c) Snapshot (d) Exit Application (e) Play (f) Record.

3.1.1 Viewpoint and Voice Capture

From the user's perspective, the view reacts immediately in response to changes in the position and orientation of the display. The microphone above the display is well positioned to capture speech from the primary annotator and from discussions with colleagues who are also viewing the display. While the Record button is active, both viewpoint and voice information are saved as separate streams, ready for immediate playback.

3.1.2 Gesture Capture

When communicating ideas, users often want to indicate positions, shapes, or leave artefacts that act as a reminder where they have been. Other times, users want to make ambient, passing comments. Consequently, our annotation system supports three tools for gesture capture: pen, flashlight, and snapshot. The Pen tool and Snapshot capability allow marking gestures that consist of precise, concrete annotations. The flashlight tool exists for more ephemeral, non-marking gestures. A description of each mode follows:

Pen. We provide the ability to draw lines directly on objects in the scene, which allow for the resulting annotations to be viewable from multiple viewpoints (see Figure 7a, b). Although the Pen tool is the default, the user can rapidly toggle either Flashlight or Pen tool by selecting the appropriate button at the top of the screen (see Figure 6a, b).

Flashlight. One way to temporarily highlight areas of a physical model is by using a flashlight. A cone of light spreads as the distance increases, which allows larger or smaller areas to be indicated. In order to support non-permanent, non-marking gesture, our system simulates a flashlight placed at the tip of the finger (see Figure 7c). The 'flashlight' is itself invisible, but a spotlight appears in the location at which the user is pointing. Preliminary tests found that the flashlight was good for indicating large areas of the model, but it was often not precise enough for detailed pointing tasks. Consequently, the decision was made to indicate the precise location by a small ball appearing on the model. Since the ball is directly on the point of interest, a person reviewing a recorded annotation session can view the area of interest from different angles (§3.1.3) and still have the ball aligned and precisely indicating a position on the model. It is also possible to infer the original location of the annotator from the shape of the spotlight. When using the 'flashlight', the virtual spotlight is placed above the model one-tenth of the distance between the viewpoint and the model. This allows the user to highlight a larger area of the model when farther away, smaller regions when closer, and the sphere can specify the exact location of features when the user is extremely close.

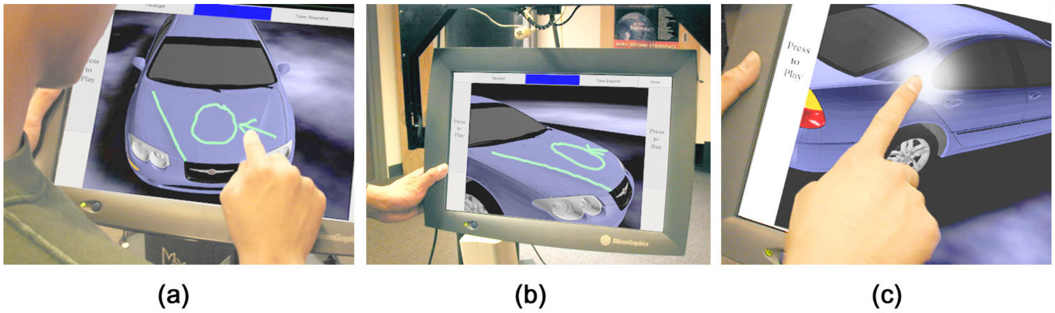


Figure 7: Marking and non-marking gesture annotations: (a) Marking gestures with the 'Pen' tool (b) Inspecting marks from an alternate view (c) The non-marking 'Flashlight' feature.

Snapshots. Snapshots in the system are flat 2D images floating in the 3D scene (see Figure 8). This differs from the approach in [69] where snapshots are windows into 3D volumes. To create a snapshot in our system, the user clicks on the 'Take snapshot' button (see Figure 6c), which causes an image of the current viewpoint to be captured and positioned in front of the camera. This also temporarily locks the viewpoint enabling the LCD panel to be used as a writing pad. The pen can be used to write directly on the snapshot while the system continues to capture voice annotations. When finished, the user presses the 'Done Snapshot' button, releasing the snapshot image.

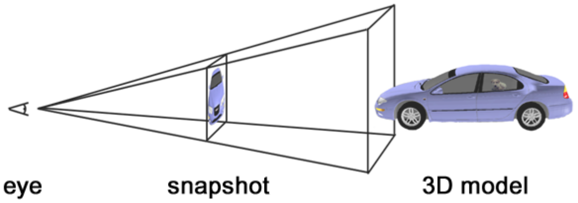


Figure 8: The snapshot feature places an image of the active view into the scene.

The snapshot capability is useful for four main reasons:

1. Although the pen can be used to draw directly on 3D objects, without snapshots there would be no mechanism for drawing where objects do not exist. This can be an especially acute problem in a sparse scene. For example, it is difficult to circle a group of atoms in a molecule (see Figure 9). The snapshot feature allows drawing 'in empty space' since the markings appear on the surface of the

snapshot.

2. As a result of the 'always responsive' interface, the viewpoint may shift while the user is drawing with the pen. This displaces the pen tip relative to the scene and can lead to deviated strokes, a phenomenon that we term *ink drift*. Normally, the viewpoint can be held steady enough to make rough marks. However, ink drift becomes a problem when precision is required. The frozen viewpoint of the snapshot helps to alleviate the ink drift problem.

3. Some viewpoints are awkward to reach and to hold. Taking a snapshot allows users to 'grab' a difficult viewpoint and reposition the display to a more comfortable viewing or writing posture.

4. Finally, drawing directly on an object can be effective in many situations, yet users may mistakenly assume the marks are being drawn on a plane in 'screen-space' rather than 'object-space'. This results in marks looking correct from the original perspective but extremely confusing and distorted from other viewpoints. For example, a perfect circle may be drawn on the ground plane from one angle, yet from another it may appear as a long thin oval. Snapshots allow a user to explicitly choose to draw in 'screen-space' and avoid this problem.

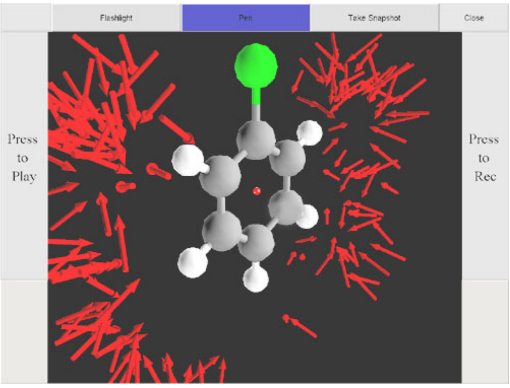


Figure 9: It is difficult to draw on sparse models, such as this 3D molecule, using only the pen tool because the geometry in the scene is relatively fragmented and marks will be discontinuous.

Animated Snapshot Transitions. Releasing a snapshot has two effects. First, it places the snapshot into the 3D scene. Secondly, to preserve visual continuity, a smooth animated transition occurs between the original viewpoint in which the snapshot was taken and the current position of the display (see Figure 10). Depending on the movement of the display while the viewpoint is frozen, it is possible to have the view suddenly move to an unexpected location. Since the abrupt change of context can be very confusing for users, we employ animated transitions to keep users oriented and prevent a visually jarring experience. This is similar to previous work animating transitions between 2D and 3D viewss [35].

Even after the viewpoint is unfrozen, marks can still be made on the snapshots. This ‘always drawable’ design minimizes the mode and tool switching that the user has to consider.

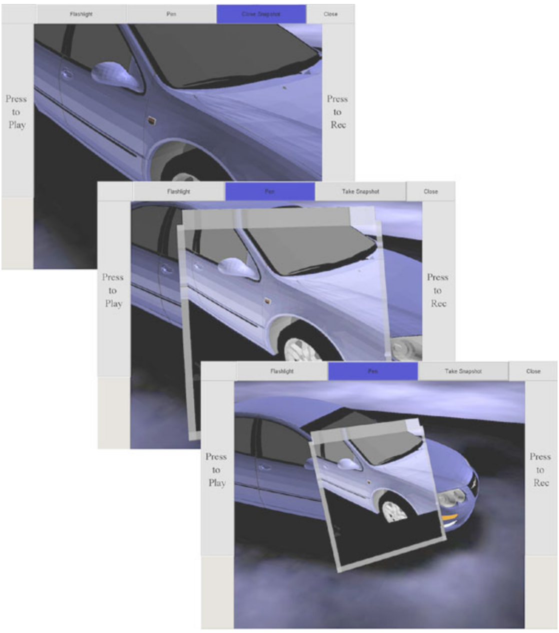


Figure 10: Animated transitions preserve user orientation after the snapshot is released.

Double-sided snapshots. In keeping with the concept of omni-viewable widgets for a volumetric display [2], the snapshot frames are double-sided, in order to show the snapshot image and annotation marks on both the front and the back of the image plane (see Figure 11a). This arrangement allows the snapshot annotations to be marked and viewed from many vantage points.

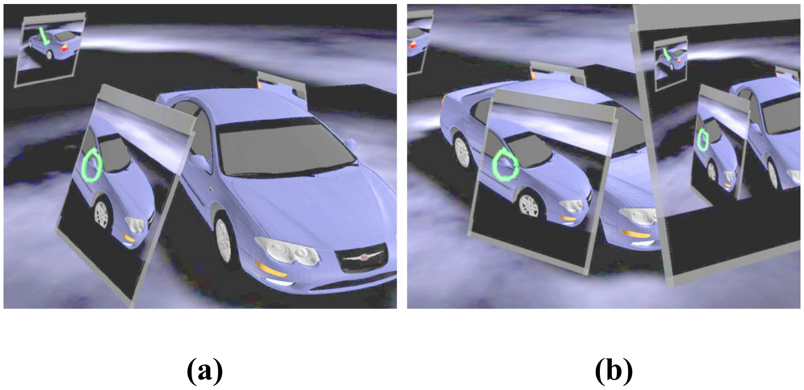


Figure 11: Snapshot features: (a) Double-sided viewing and marking (b) Snapshot layering.

Layering multiple snapshots. A convenient by-product of spatially embedding snapshots is the ability to layer multiple snapshots (see Figure 11b). Users can take a snapshot of the 3D object with previously created snapshots in view. This allows users to capture snapshots-within-snapshots to comment on existing annotations within the shared 3D scene by referencing previous snapshots[9]. We have noticed that there is a natural layering in which the earliest snapshots are positioned closest to the center of the scene while new snapshots are positioned further away.

Snapshot Titlebars. Populating the 3D scene with many snapshots can clutter and obscure the 3D object. To counteract this effect, we have added the ability to hide the snapshot images. Clicking on the titlebar of a snapshot toggles the visibility of the image. The titlebar is clickable from both the front and back, and is drawn semi-transparently to reduce the problem of obscuring the model. Lastly, while a snapshot is hidden, the position, size and orientation of the residual titlebar provide the user with an awareness of the scope of the hidden annotation (see Figure 12).

[9] Note this is commenting on existing annotations is possible by drawing directly on the previous snapshots with the Pen tool, but layering snapshots allows the user to more easily reference multiple snapshots by encapsulating the comments in the 'meta' snapshot.

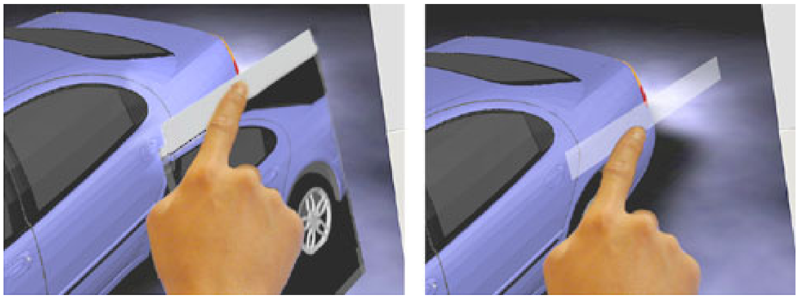


Figure 12: Toggling snapshot visibility using the titlebar.

3.1.3 Session Management and Workflow

Recording. To begin an annotation session, the user presses the Record button (see Figure 3f). This activates the capture of all three data streams: viewpoint, voice and gesture. The Record button turns red to indicate that data is being captured, and clicking on the button a second time will end the annotation session.

Playback. To replay the last annotation session, the user may click on the Play button located on the left side of the display (see Figure 6e) and the recorded annotation session is then played back in a separate window (Figure 13). Note that the 3D object and annotation data is also presented 'live' in the main window. This alternative real-time view of the annotation session allows the user to have a user-controlled viewpoint during playback. This is useful, for example, if the recording is pointing out a flaw with a part of the model and the user wishes to get a closer look or view the flaw from a different angle. Alternative 3D views have previously been offered for navigation [76] but not for live annotations.

Saving. From the playback window, a user can choose to save the annotation session. A representative thumbnail image of the annotation session is then generated and used as part of the workflow, which is described below.

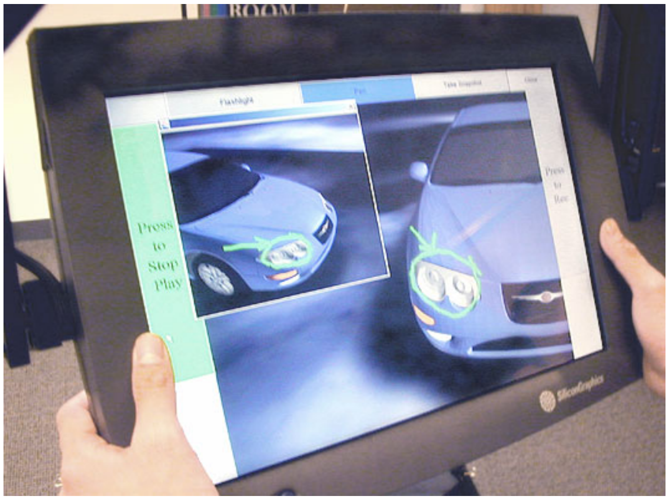
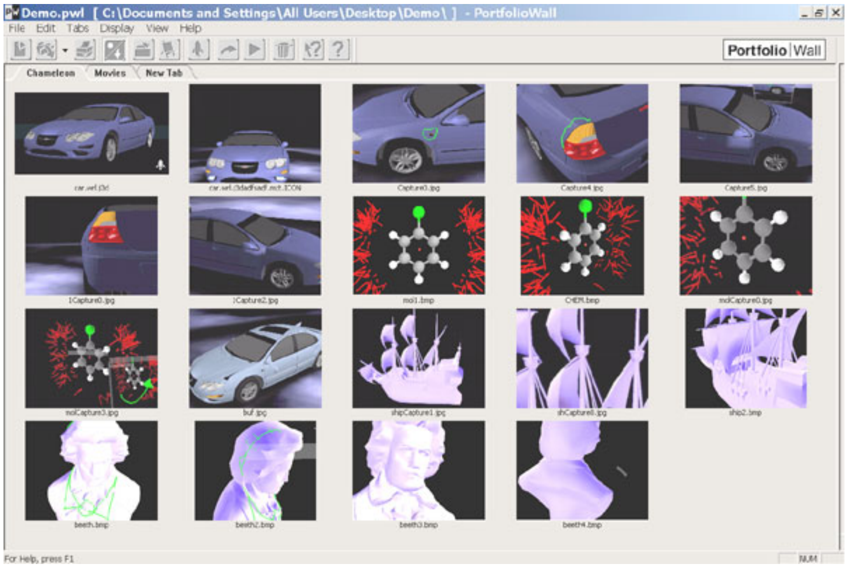


Figure 13: A separate playback window appears during playback. The original window is still visible in the background and responds to user feedback.

Workflow. To begin a design review, the user can select a 3D object from a 2D image browser (see Figure 14) running on the Boom Chameleon. Clicking on an image will launch the annotation application with an appropriate 3D model. Saved annotation sessions will also appear as thumbnails in the browser and, if selected, will playback the session. Users can also transmit an annotation session to their colleagues by sending the data as an e-mail attachment.


Demo.pwl [C:\Documents and Settings\All Users\Desktop\Demo\ ] - PortfolioWall

For Help, press F1

Figure 14: A 2D browser allows the user to access 3D models and annotations through thumbnails.

3.2 Design Issues

Some issues and considerations were encountered while iteratively designing and testing the system. This section discusses the growth of research through difficulties. It then describes features of the annotation interface, including widget placement and snapshot rationale. Model scale, position and sparsity are also considered. Concluding the chapter is a section on the contexts of use, which examines the different roles taken by the system in different situations. Strengths and weaknesses of the system are discussed in the context of each role.

3.2.1 The Role of Failure and Redesign

> "… the form of made things is always subject to change in response to their real or perceived shortcomings, their failures to function properly. This principle governs all invention, innovation, and ingenuity; it is what drives all inventors, innovators, and engineers … Since nothing is perfect, and indeed, since even our ideas of perfection are not static, everything is subject to change over time ..."
>
> -- Henry Petroski, *The Evolution of Useful Things* (1992)

It may be instructive to show how difficulties and failures guided the formation of novel ideas and techniques during the course of our research. In this section we run through one example of how many techniques arose from the investigation of a single problem.

Drawing on a sparse scene. Drawing on a sparse scene presents a problem for the pen tool. Because the pen tool creates marks directly on scene geometry, with it we are not able to draw in empty space.

One approach to solve this problem was to use a threshold value when creating the pen marks. If objects were close together, the marks can 'jump' between objects and a line would be drawn in the empty space between the objects. However, this solution presented a problem. A reasonable threshold for what was considered 'close' endpoints was highly dependent on the nature of the model. Even with a reasonable value, unwanted connections often appeared.

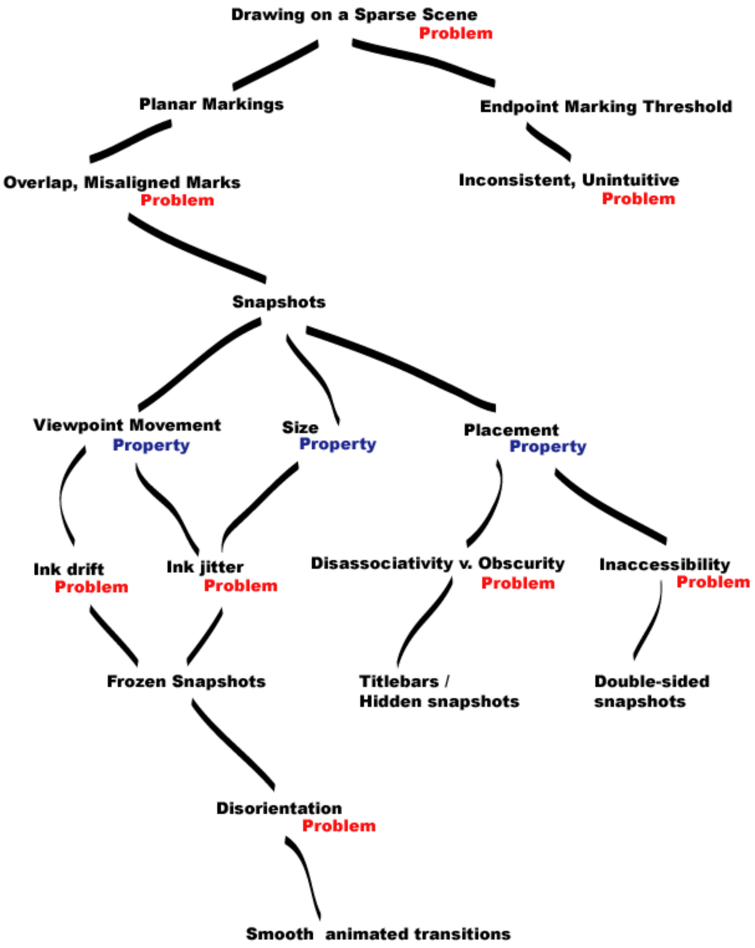


Figure 15: One example of how a series of difficulties helped shape our research resulting in novel solutions and techniques. Starting from the problem of creating marks in the empty space of models, we negotiate four tiers of problem, solution and refinement.

Planar marks. We then considered creating marks on invisible planar surfaces. However, if many unrelated marks were drawn close together, superposition of marks due to overlap may result in confusion.

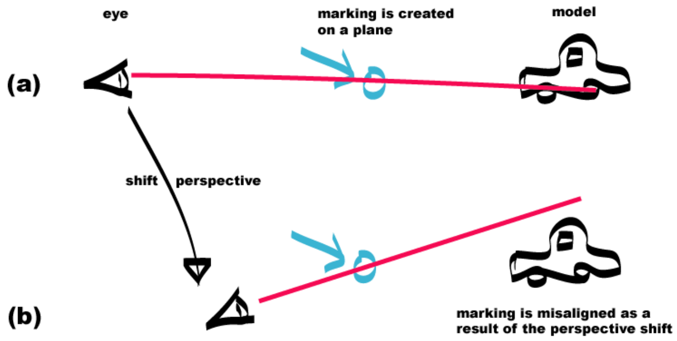

eye
marking is created on a plane
model
(a)
shift perspective
(b)
marking is misaligned as a result of the perspective shift

Figure 16: Planar marks on invisible surfaces (a) Marks created on a plane between the viewpoint and the model are aligned from the perspective of the original viewpoint; (b) however, when the viewing angle changes, the marks are no longer aligned with the model.

Misalignment of the marks was also a concern. Shifting the perspective would cause difficulties when associating and aligning the marks with the model (see Figure 16). A viewpoint that 'snaps' to the plane when in close proximity would fix the alignment problem, but it also has the potential to cause practical problems during navigation in a heavily annotated space.

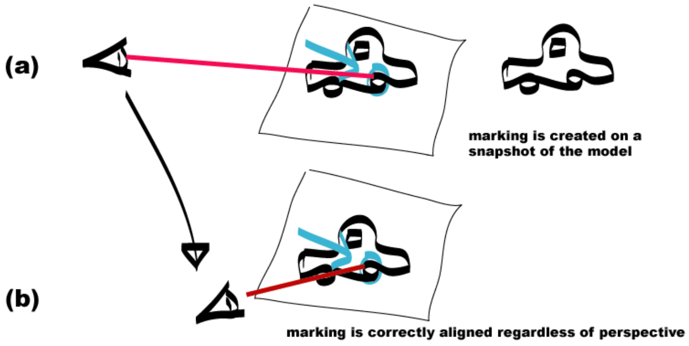

(a)
marking is created on a snapshot of the model
(b)
marking is correctly aligned regardless of perspective

Figure 17: Snapshots. (a) Marks created on an opaque picture of the model are embedded in the scene. (b) Since the marks are drawn directly on the snapshot, the marks are correctly aligned, regardless of any change in perspective.

Snapshots. Replacing the planes with opaque pictures embedded in the scene solved the alignment problem because the pictures are always in alignment with the marks that are placed on them (see Figure 17). This contrasts with marks 'floating in space' where the alignment depends on the viewpoint. The opacity of the snapshots also eliminated unintended superposition of sets of proximal, but unrelated marks.

The size, position of the snapshot feature and movement of the device gave rise to several problems and solutions.

Snapshot Size, Device Movement -> Ink Jitter, Ink Drift. The snapshots were designed to encompass the entire field of view. This means that if a snapshot is positioned close to the viewpoint, the absolute size needed to fill the field of view was smaller than if it was positioned farther away. With the snapshot very close to the viewpoint, the scale of the picture and marks are small. As a result, the tiny, normally imperceptible movements of the display (due to the precision errors of the sensors and hand tremors) mean that during drawing, the snapshot may suddenly shift relative to the viewpoint. The marks being drawn on the snapshot consequently appear ragged and discontinuous. This problem of *ink jitter* is still present but insignificant when snapshots are placed far away from the viewpoint and have a larger scale, but the problem is exacerbated when the snapshot has a small scale. Another related problem was *ink drift*, first mentioned in §3.1.2. This has the same effect as ink jitter, but results from large-scale movements of the device, for example, the act of using the touchpad will move the device. Ink drift affects both small and large scale snapshots.

Ink Jitter, Ink Drift -> Frozen snapshots. Microscopic ink jitter and the analogous macroscopic ink drift were both eliminated by freezing the viewpoint, so that the user is drawing on a static snapshot.

Frozen snapshots -> Smooth animated transitions. Unfortunately, the freezing may cause user disorientation when the viewpoint is unfrozen. If the displacement of the display between freezing and unfreezing is large, the viewpoint can move dramatically and instantaneously. To combat this problem, we implemented smooth animated transitions between pre-freeze and post-freeze positions.

Snapshot Placement-> Disassociativity, Obscurity, Accessibility. A tradeoff arising from the position of the snapshots is disassociativity versus obscurity. If snapshots were positioned close to the model, they will be associated with the relevant parts of the model. However, this comes at a cost of obscuring the model. Snapshots positioned far away from the model risk tenuous association with the subject of interest, but are less likely to obscure the model or other snapshots.

Disassociativity, Obscurity -> Titlebar. The tradeoff of obscuring the model with opaque snapshots was solved by allowing the pictures to be hidden. To do this we used semi-transparent titlebars to toggle visibility. The titlebars also hint at the position and orientation of the associated snapshot when the snapshot is hidden.

Accessbility -> Double-sided snapshots. As snapshots are placed in different orientations, it may become tedious to view them all from near the original location. Double-sided snapshots allow greater accessibility of the information.

From the problem of writing on objects to the prevention of disorientation by a smooth animated transition, difficulties played an important role in the shaping of our research direction and resulted in many novel ideas and techniques.

3.2.2 Interface Considerations

Intimate User Interface. The interactive view is omnipresent and consistent. Because the interactivity and sensation of responsiveness is never[10] interrupted, the physical/virtual spatial congruency is continuously maintained. This characteristic of the system allows the maintenance of what we term an *intimate user interface:* a consistent and continuous dialogue of interaction with the user that is never jarring or unexpected.

The term 'intimacy' when used in relation to devices is not new; Moore introduced the term "control intimacy" [58] in 1988 with respect to musical instruments:

> "The best musical instruments are ones whose control systems exhibit an important quality that I call "intimacy". Control intimacy determines the match between the variety of musically desirable sounds produced and the psychophysiological capabilities of a practised performer."

Fels [25] uses a similar definition of intimacy with devices:

> "Intimacy is defined as the subjective match between the behaviour of a device and the operation of that device" he goes on to describe in [24]: "When a person has a high degree of intimacy with a device they can communicate ideas and emotions effectively through the device as if it were an extension of themselves."

Our use of the term 'intimate' has a similar connotation, but a dissimilar definition. Our definition of an intimate interface refers to a constant, yet comfortable real-time reactivity between the user and the interface, which affords$_{(G)}$ a consistent and continuous dialogue of interaction. The interactive view of the annotation application is an example of an intimate interface, and it affords$_{(N)}$ a strong sense of user control. We believe that this is one reason why users seem immediately comfortable with using the Boom Chameleon.

[10] An exception is the freezing of the viewpoint during a snapshot. However, this freezing of responsiveness can cause disorientation.

Button Placement. In the current annotation graphical interface, buttons on the top of the screen indicate the flashlight gesture, pen marking, snapshot mode, and the button at the upper right corner exits the application. The two buttons running down the sides of the display facilitates grasping anywhere along the side, and are used to record and review annotations. Our initial design had button that ran the entire length of the screen, however, we found some users accidentally triggered the recording or playback when quickly grabbing the display by the bottom corners. As a result, there is a small inactive region at the bottom of each side button.

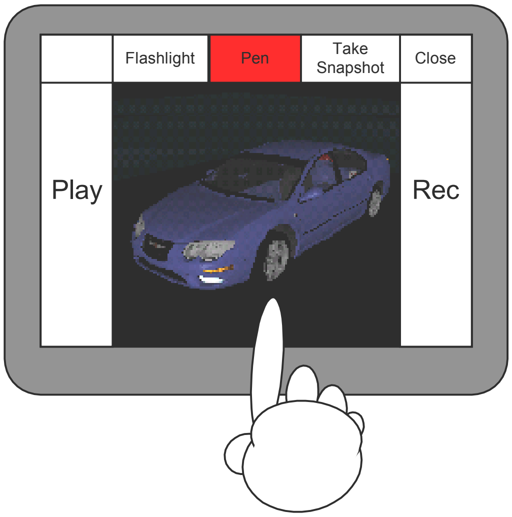


Figure 18: Widget layout includes inactive spaces near the bottom corners of the screen to prevent users from accidentally triggering the recording or playback mechanisms.

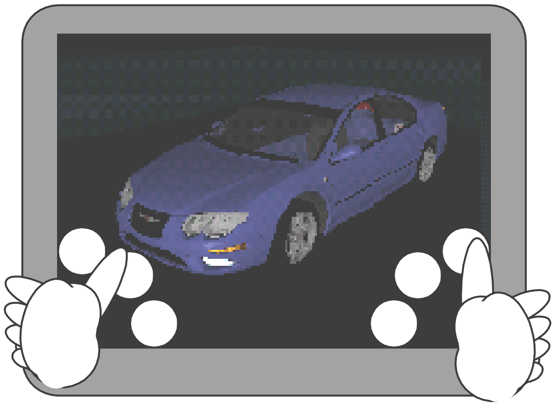


Figure 19: A radial thumb-button layout was considered, but was not used.

An alternative design employed radial buttons for the thumbs. From each gripping position, minimal movement would be necessary to reach any button. Although one advantage radial buttons have over the current design is the greater area available for viewing the model with less area consumed by the buttons, this concept was relinquished in favour of the current design, which allows for a greater number of hand positions and straightforward grouping of buttons.

Quality of Snapshot. Preliminary testing found that users tended to forget that they were in snapshot mode. In spite efforts to signal state with the colour of the snapshot button, users often assumed a mark being made on the model in free-form mode, when in fact the mark was being placed on the snapshot in a frozen view. Users only discovered their mistake after they tried to navigate away from the snapshot and were unable to move the frozen viewpoint. After further investigation, it was found that the visual image quality of` the model was identical both in and out of snapshot mode, which may have confused the users. Users did not experience this "state-forgetfulness" problem less often when a low quality bitmap was used for each snapshot. Using low-quality bitmaps for the snapshot texture produces a grainy, yet reasonable picture, which contrasts with the "normal" operation and constantly reminds the user that they are currently looking at a snapshot. One might imagine using a non-latching button to hold snapshots would be another way to reduce problems through muscular tension and closure [11].

Placement of Snapshots. One issue with the snapshots is the placement and size of the snapshots in the scene. If placed too close to the model, the snapshots may obscure or even intersect the model. If placed too far away, the relation between snapshot and model may be lost. Our current implementation uses a fixed distance, however, a variable distance, perhaps one third of the distance between camera and model, may be more appropriate. The best practices for these snapshots remain to be seen.

Control mappings and Transfer functions. Control mapping is the type of mapping applied to the device controlling the interaction, such as positional control, rate control, or acceleration control. Within each mapping, a transfer function defines the relationship between the input and the output. The Boom Chameleon system uses a position, or *pure* *gain* (1:1) mapping [85], and our transfer function is a simple linear relation. Although using non-linear transfer functions may allow the user to reach areas that are unreachable using linear functions, we decided to use a one-to-one mapping with simple linear transfer functions.

3.2.3 Model Considerations

The metaphor and interaction characteristics for navigation and viewpoint are usually determined by the browsing mechanism; however, the information being viewed plays a surprisingly important role when using the Boom Chameleon annotation system. The scale, position, and type of model to be annotated significantly affect the interaction with the system.

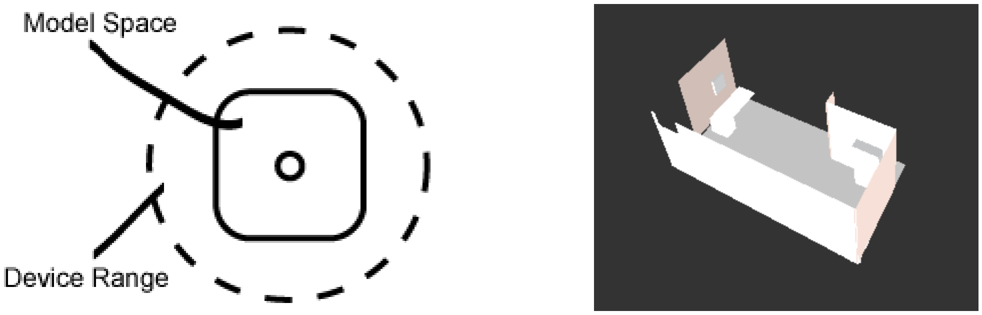


Figure 20: Small models are more conducive to an exocentric viewpoint.

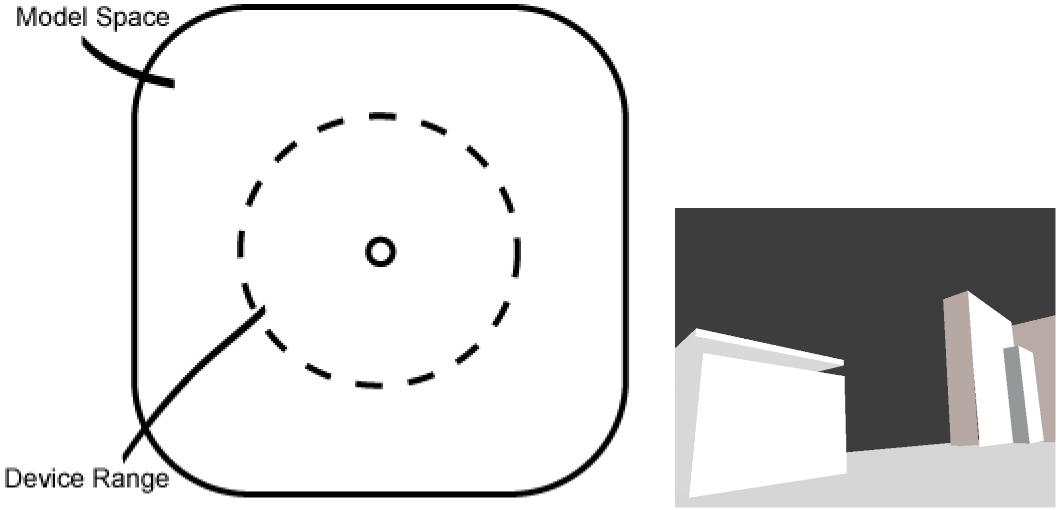


Figure 21: Large models are more conducive to an egocentric viewpoint.

Scale. Models that are small in relation to the range of the Boom Chameleon are conducive to exocentric viewing (see Figure 20). However, when scaled to a large size, these same models can be comfortably viewed in egocentric fashion, with no change in the underlying mechanics of interaction (see Figure 21). For a description of the differences that are afforded$_{(G,N)}$ by egocentric and exocentric viewpoints, please see Appendix A.

Position. We have found that the affordances$_{(G,N)}$ are not only affected by scaling, but embedding the model at different positions also results in differing affordances$_{(G,N)}$. For example, in our tests with a fixed model, some users noted that they would not be able to reach some desired viewpoints because of the physical limitations of not being able to fly up and directly over the model. Still others desired to position the view at the centre of the model, but could not because of the centralized physical pole of the Boom Chameleon.

One solution to the unreachable viewpoint obstacle would be to implement a clutching mechanism for manipulating the viewpoint; however, this would break the spatial anchoring of the model in physical coordinates and may cause difficulties due to lack of cognitive-spatial consistency. One solution to allow the viewpoint to traverse the centre of the model would be to have a model shifted, so the spatial mapping is a shifted one (see Figure 22). Another solution would be to mount the armature to the ceiling.

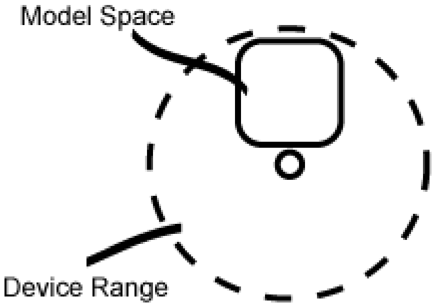


Figure 22: A model placed off-center allows the display/viewpoint to be passed through the center of the model.

Model Sparsity. While working with different types of models, we found noticeable differences in our annotation between contiguous "solid" structures (see Figure 23), and sparse "space-filled" structures. The pen tool writes directly on the 3D geometry of objects, so markings can easily be made on the models. However, with a sparse model (Figure 24) there is more empty space than there are object surfaces in the scene. Writing strokes that extend beyond the model is important for users to express extensions, additions, and many other intentions during annotation. Although we could decide to deposit pen strokes into the empty space close to the geometry, alignment problems arising from different viewpoints would present difficulties. As a response to the problem of empty-space writing, snapshots were created. Snapshots capture the current viewpoint of the scene as a writable, homogeneous flat surface and we are able to write onto the empty spaces without causing perspective problems (see Figure 25). Furthermore, the snapshot position and orientation allow users to deduce the perspective from which the snapshot was originally made (see §3.1.2).

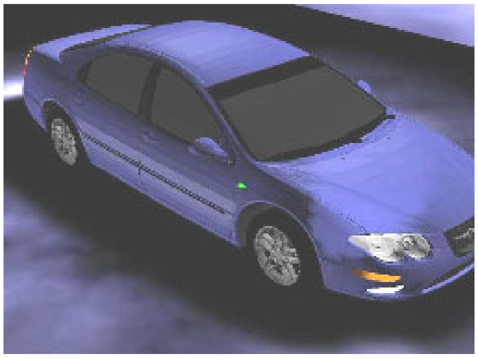


Figure 23: It is easy to write on a solid model.

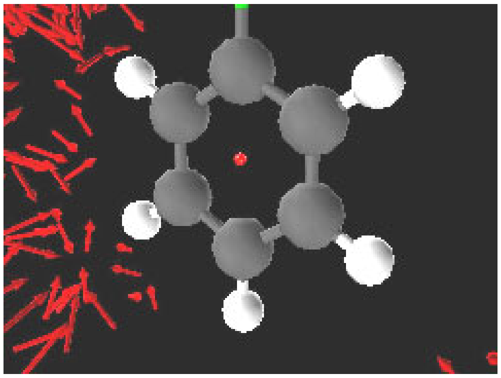


Figure 24: A sparse model makes it difficult to draw contiguous strokes with the marking pen tool, since the marks can only be projected onto geometry and not empty space.

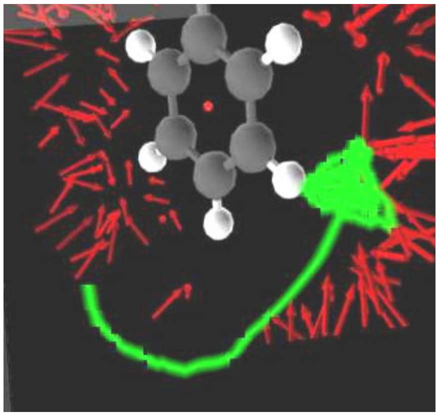


Figure 25: A Snapshot allows marks to be drawn in "empty space", without alignment problems arising from viewing annotations from different angles.

3.3 Contexts of Use

> *Thoughts exchanged by one and another are not the same in one room as in another.*
>
> - architect Louis I. Kahn

The Boom Chameleon system is built using a specialized device and not an all-purpose tool, yet the role of the system may change simply by being placed in different environments or situations. In fact, there are some who assert that the physical, social, and machine-relative contexts are important in defining the use of a device [13]. In this section, we examine differing roles in each of the task, social, and machine contexts.

3.3.1 Task Context

When a device is used for different tasks, the characteristics of the device are used in different ways. This section presents four task-based scenarios involving browsing, presentation, and playback. The role taken by the Boom Chameleon system is described in each situation and the strengths and weaknesses of the system are compared to conventional approaches.

Browsing, a clay model equivalent: In the browsing environment, the Boom Chameleon may act as the functional equivalent of a clay model by serving as a showcase. Passers-by may casually browse a model or even converse with others while using the model as the talking-point for discussion.

In this scenario, the Boom Chameleon may function as an "always-on" display that is inviting to exploration by way of its low interaction requirements. When compared to traditional virtual reality browsing systems, the Boom Chameleon is more inviting to 'passing interest' because it does not necessitate the physical engagement required by sitting down at a desk, donning gloves or wearing goggles. The Boom Chameleon needs neither physically committing action nor encumbering equipment, both of which serve as barriers to the concept of "walk up and use". Furthermore, group interaction may benefit from this same phenomenon, since the gesture of handing over the display to another person encounters less expense when compared to having each person sit down at a desk.

Both the Boom Chameleon and traditional clay and fibreglass models allow large groups of people to respectively look at the duplicate display signal or physical model. However, the physical model is more expensive but allows for touch, and individuals may simultaneously and independently control their own viewpoint. This capability for many simultaneous, yet independently controlled viewpoints is not a capability supported by most digital viewing systems, including the Boom Chameleon. Therefore physical models are better suited for use if tactile feedback or multiple independent viewpoints are needed.

Yet, when only a single viewpoint is required, the Boom Chameleon brings the flexibility of digital media to the fore: the Boom Chameleon system can load virtually any 3D model in contrast to the single-use physical clay, cork or fibreglass prototypes. However, if only one model is needed then a small physical prototype maybe more economical to build than the current implementation of the Boom Chameleon. Another disadvantage of the Boom Chameleon is that initially, compared to physical media, the interface may be unfamiliar and therefore more daunting.

Presentation, a Digital Podium: The Boom Chameleon may also be used to deliver a presentation. Building on the browsing scenario, a speaker presents information using Boom Chameleon as an interactive and semi-movable podium. The presenter is afforded$_{(G)}$ exquisite control over the viewpoint of 3D models while the result is reflected on a large adjunct screen for the audience to see. The device acts as a presentation tool, a role that takes advantage of the Boom Chameleon's strength in navigation and allowing the user to concentrate on the speech instead of worrying about working with the model.

In current practice, a naïve presenter would need an engineer or pre-recorded movements to drive the three-dimensional viewpoint. The advantage of the Boom Chameleon in this situation is that it allows the presenter to perform spontaneous, improvised explorations without significant cognitive overhead. As a result of these improvements, the capacity for spontaneity should increase, as should the capacity for using many different parts of the model. One disadvantage is that the speaker may not be able to maintain eye contact with the audience while using the Boom Chameleon.

Capture and Playback: This scenario uses the system to send the annotation session to others and to have the session reviewed asynchronously, but in real-time. In this case the Boom Chameleon kiosk is used as a data source in an extended workflow.

Although the annotation sessions can be transmitted to and played back on any standard workstation, the annotation facility highlights one of its features when annotations are reviewed directly on the Boom Chameleon. The original recorded viewpoint is shown in parallel with an interactive viewpoint, allowing the two views to be compared. The user may probe the model with the interactive view during playback of the recorded session so that the two viewpoints can be compared and contrasted even as gestures and markings are being re-created. Viewing the model and associated markings from a new perspective may better elucidate the comments given in the annotation session. Multiple reviewers are able to inspect and comment on the design knowing that most important aspects, save perhaps body language, are being captured for future review by the recording facilities.

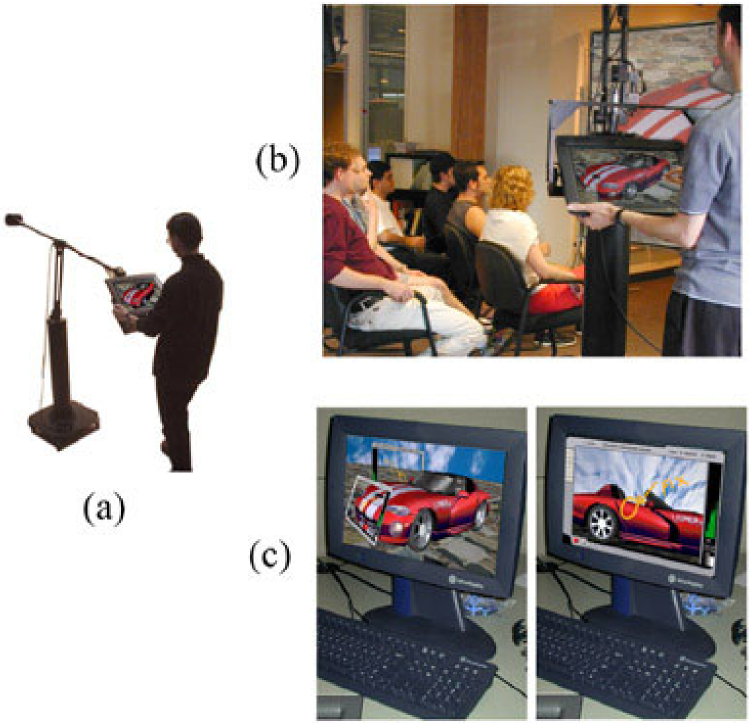


Figure 26: Task context. The system adopts different roles depending on the task (a) Browsing on the chameleon; (b) A 3D presentation podium; (c) Playback on a standard workstation terminal.

Using the Boom Chameleon in the Browsing, Presentation and Capture/Playback scenarios result in improvements to the conventional design review that address the viewpoint, shared view and information capture problem.

3.3.2 Social and Machine Context

While the previous section considered the Boom Chameleon in a more standalone task-centered context, this next section describes the role taken by the device when placed in the social and machine contexts.

Social Context. When we consider the Boom Chameleon in a social role it, may act as a common or shared virtual space that allows its users to gain a better understanding of the phenomena of interest. For example, managers may further their knowledge of the design intent by browsing, designers to enhance their understanding of what others perceive of their work through the resulting annotations. Dourish and Bly introduce an early example of this type of awareness [19] with a system called "Portholes". Portholes facilitate the social function of awareness to see "what's going on" in distributed work groups. Dourish and Bellotti examine the impact of shared awareness in more detail in [20]. Buxton [10] introduces the term *awareness server* when describing systems such as Portholes.

When taken in the social context, our system may act as an awareness server. As models are created and new annotations are composed, they automatically appear in the browser displayed on the Boom Chameleon (§3.1.3). This puts the work into the shared periphery and helps support the "background social ecology" [10] of the workflow.

The following scenario illustrates how the system promotes awareness by making models and annotations readily accessible to the user. One may be walking by the Boom Chameleon and notice a design being cycled on the display. Later, one may walk up and start browsing for the particularly eye-catching design that was noticed earlier. Perhaps the design is found, but something else, which was shown alongside that design, seems more interesting and is examined in detail. This scenario illustrates peripheral awareness and serendipity, contrasting with the scenario where information is stored in a location that is not immediately accessible or visible. With the 'hidden' information, there is no way for the user to know if interesting models or annotations were created or changed other than to explicitly search for them. This approach lacks opportunities for peripheral and serendipitous awareness.

Appendix B shows how awareness servers can make series of tasks easier to perform through the interaction of the task and social contexts.

Machine Context. To understand how a group of machines, each performing a different task, can affect the role of a particular device in the group we should first understand why we have a different machine for each task. To do this, we review the interface hierarchy introduced in §2.1.2.1.

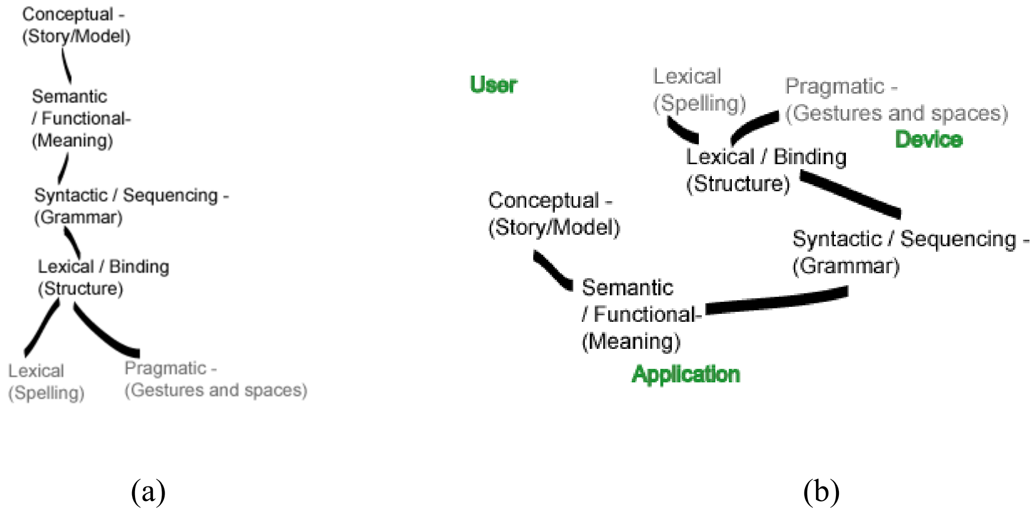


Figure 27: (a) The Foley et al. [30] / Buxton [9] hierarchy (introduced in §2.1.2.1); (b) an alternative layout of the hierarchy emphasizes the dependence of each hierarchy layer relative to the user, application, and device. Dependence is represented here by proximity.

The taxonomy shown in Figure 5 (and Figure 27a) is a simple hierarchy that suggests a separable layering of concerns. Yet, if we use a different device for the same task, the effect on the hierarchy is not immediately clear. The device strongly effects the interaction [9], but the semantic meaning of an action is very dependent on the application, and the concept of the system depends on the user. This suggests organizing the layout of the hierarchy to highlight the dependence on the user, application and device (see Figure 27b). We can use this layout as a tool to help analyse the impact on interaction when we alter any part of the user, application, or device. See Appendix C for a detailed explanation of the figure.

We now see that changing the device impacts most of the hierarchy. We proved in §2.1.2.1 that ease of use follows from alignment of the hierarchy, and we assert that misalignment correspondingly makes tasks more difficult. If we try to do many different tasks on one machine, it will likely decrease the alignment of the hierarchy, making the task more difficult (see Appendix D).

On the other hand, a strong specific device may be used with applications and tasks that tailor to its unique lexical and pragmatic characteristics. Exploiting this synergy can lead to strong resonance throughout the hierarchy, making it easier to perform tasks as asserted in §2.1.2.1. Strong specific devices working together may each address a small aspect of a workflow, and each machine can excel at the specific sub-task. When viewed as part of the group of devices, a strong specific device can play an integral role in the machine context. The advantage using many strong specific devices is qualitatively presented and introduced as the "Net Benefit" in [13].

The following example illustrates how the roles of the Boom Chameleon system, standard workstation, flat-panel display, and immersive display are shaped by each other: The Boom Chameleon system excels at creating annotations, and playing annotation sessions on the Boom Chameleon itself is conducive to understanding feedback and especially comments about space, since an alternative view of the annotations are available during playback (see §3.1.3). Yet if one wanted to modify the model in response to the remarks, a standard workstation is better suited to do so. If a large format 2D display is available, then it might be used to browse through the 2D thumbnails of models and sketches. We are also more likely to use a CAVE™ or other immersive display to explore the model from an egocentric viewpoint, rather than using the Boom Chameleon (§3.2.3). Although in theory, any device could do everything that the other devices do, each device has its own strengths, which determine its use when taken in context of the other devices.

The task, social, and machine contexts overlap and interact to influence the role adopted by our system.

3.4 System Implementation

This section describes the hardware, software and event-based architecture that process the interaction of the annotation application.

3.4.1 Hardware

The Boom Chameleon is a specialized input/output device that is a modified version of the Fakespace BOOM3C™. The BOOM3C consists of a pair of stereo goggles mounted on a position and orientation-sensing armature [23]. With the Boom Chameleon, the goggles were replaced by a 15" (16:9 aspect ratio) Silicon Graphics LCD panel (see Figure 1). The armature was then re-balanced in order to have the display float weightlessly in space. With this configuration, the display has a 10-foot spherical range of motion around the center of the device.

To facilitate touch input a customized Elo™ [21] touch screen was overlaid onto the LCD, allowing the hands to operate two modes of input simultaneously. This means that one hand may control the position of the display while the other utilizes the touch screen. To record voice, a podium microphone was attached to the top of the display frame. Standard computer speakers were used for sound output.

The current system hardware uses an Intel Pentium III 500 MHz dual processor running Microsoft Windows2000 with a NVIDIA GeForce2™ graphics card.

3.4.2 Software

The annotation application is implemented in Java™ and the Java3D™ API with an event-based architecture to collect and record information. VRML files are read into the scene-graph courtesy of a loader provided by Sun Microsystems. A device driver produces navigation information and the system synthesizes the various angles from the joints of the armature into a position and orientation value event.

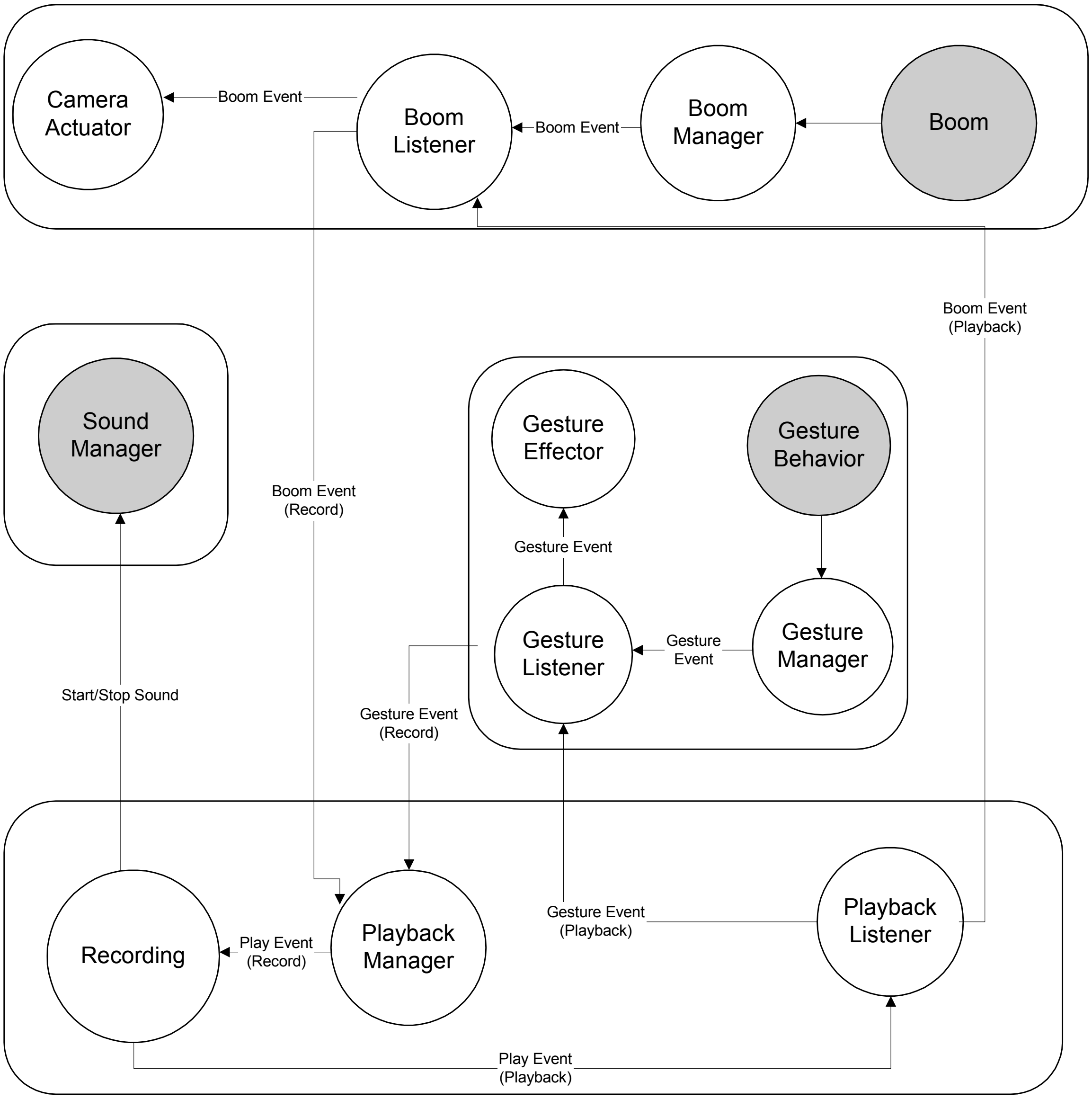
Touch input is received as a standard mouse event. Depending on the state of the interface and location of the input, the event may be processed as a button push, pen drawing, or a highlighting gesture. Drawing on objects in the scene is accomplished by attaching Java3D™ behaviors to the scene graph to detect the picking intersection with scene geometry. When appropriate mouse drags occur, the system performs continuous picking to determine the intersection. This picking information is then used to place annotation marks that follow the scene geometry, but are slightly offset above the geometry to make them visible to the user.

Capturing the image in the main window as a JPEG image is the first step in creating snapshots. The image is used as a texture on both sides of a new flat polygonal surface to be placed into the scene. The polygons and texture are placed in the scene graph under one node for each snapshot. This convention allows snapshots to be easily hidden or displayed by simply detaching or reattaching the appropriate node.

Audio is recorded and played by the standard Windows and Java™ sound application programming interfaces respectively.

During recording, all events are time stamped and saved for later use. However, these events are also processed immediately, regardless of recording state, in order to update the display with the appropriate changes in the viewpoint or annotation of the scene.

During playback, a playback window appears as a view on the 3D scene, while the main window remains active in the background. Recorded events are sent to both the playback and main windows but interpreted in different ways. In the playback window, all events are interpreted, and thus all session viewpoint movements and annotations reappear in the window. On the other hand, the main viewing window ignores the viewpoint information being replayed and the main viewpoint remains controlled by the position of the display. Other recorded events such as flashlight pointing or drawing do occur in both the windows. The alternate, user-controlled view of the main window allows the user to view the real-time creation of annotations from viewpoints that differ from the original viewpoint being shown the playback window.



Arcs containing (Record) are only active during recording. Similarly, arcs with (Playback) are only active during playback.

Notice that arcs must go through Listener Objects to cross group boundaries or to reach effectors. Disabling the Listeners will disable the respective group functionality.

● indicates user input

Figure 28: The architecture diagram for the event-based annotation application.

Chapter Four:

System Evaluation

The web of our life is of a mingled yarn, good and ill together.

- WILLIAM SHAKESPEARE

This chapter describes our evaluations of both the Boom Chameleon device and the annotation application. The first section describes the evaluation of the view-only device. The second section presents the setup, procedure and the results of an informal evaluation of the annotation-enabled Boom Chameleon.

4.1 View-only Evaluation

The first version of the Boom Chameleon was built three years ago at Alias|Wavefront by George Fitzmaurice. This initial version was a "view only" device with no touch screen or microphone, and limited users to moving around a 3D scene without annotation. Although this device has never been formally introduced in a publication, there has been extensive experience gained through hands-on demonstrations of this initial version, as it has been operated by hundreds of users at SIGGRAPH'99 as part of the trade floor exhibits. There have also been hundreds of hands-on demos conducted at the Alias|Wavefront research laboratory with users ranging from experienced 3D computer graphics artists to technology-shy corporate executives.

Through our own informal evaluations, we have observed that every one of the users immediately understand the navigation metaphor and are able to control their navigation within seconds. The single instance of a user "becoming lost" occurred when a user moved below the virtual floor in a model and encountered an entirely black scene. We have also observed a few users initially confused when navigation started inside a model of a car without a modeled interior. Such an interior presents an unfamiliar and confusing display, but is quickly remedied as the user moves by chance to the outside of the car. One may venture to say that, in this instance, the main cause of confusion is the model and not the navigation metaphor.

One interesting and often repeated social observation we term the *herding effect*. The herding effect occurs when small groups of people attempt to observe the screen of the Boom Chameleon while someone is attempting to navigate through a model. This results in a "follow the leader" situation where the group flocks back and forth to look over the shoulder of the user. This effect is especially pronounced when the user is hierarchically superior to others in the group. Since the herding effect results in a clumsy and ineffective method of observation, the situation has been improved by providing a fixed duplicate of the display in close vicinity to the Boom Chameleon. This setup successfully counters the effect and suggests the Boom Chameleon's role as a 3D podium. Such a scenario was described in §3.2.3.2, whereby a presenter may drive the interaction while an audience watches on a separate fixed display.

It was also observed that some users realize the limitations of the viewpoint control technique. For example, one user asked how he could "fly above the car to look straight down from the top", which could be addressed using clutching (§3.2.3), or non-linear gain mappings (§3.2.2).

4.2 Annotation Evaluation

The Boom Chameleon was augmented with a touch screen, microphone and our specially designed annotation software. Using this annotation-enabled version, we performed two sets of user evaluations. First, twenty-three people operated and commented on the system, resulting in modifications to our application, such as additions to the annotation feature-set (§3.2.1) and changes to the widget layout (§3.2.2). After these modifications were implemented, users had little trouble learning and operating the refined interface.

In accordance with our goal surveying the design space, we conducted an informal study focused on how users simultaneously use speech, viewpoint control, drawing, and pointing in annotation. To study this behaviour we recruited six subjects and asked them to perform a task with the Boom Chameleon. None of the subjects had previously had experience with the annotation system.

4.2.1 Setup

We first presented the subjects with an operational overview of the system. The annotation functionality was demonstrated and the user was given a short time to familiarize themselves with the system by annotating on top of a standard car model.

After this short acquaintance period, we revealed that their task involved searching for, and reporting on, major defects that were introduced into an altered model. The car they were to inspect had been vandalized resulting in “some severe and dramatically strange defects – probably the work of an angry wizard”. Subjects were asked to play the role of an insurance claim adjuster and annotate both the defects and possible repairs.

To keep task times short, we added six very obvious defects to the car model ranging from a sagging driver’s mirror, a crooked sunroof, an extra tail fin, an oddly rotated tire, and some large dents in the body of the car (see Figure 29).

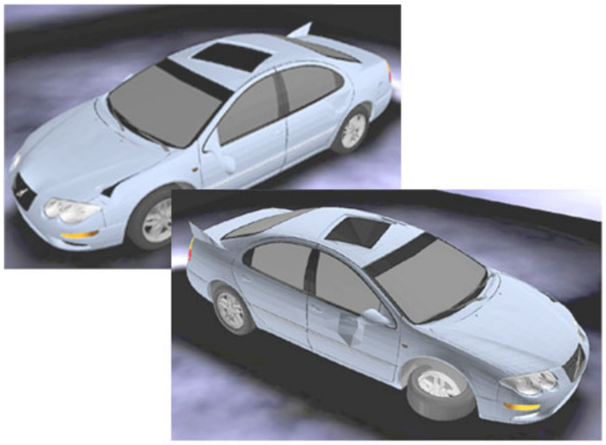


Figure 29: The informal user test involved six very obvious defects to the car model, including a sagging driver’s mirror, a crooked sunroof, an extra tail fin, an oddly rotated tire, and some large dents in the body of the car.

The video signal from the Boom Chameleon was duplicated and displayed on a separate monitor that was filmed by a standard video camera. The microphone from the Boom

Chameleon was also fed into the camera input. During the annotation session, the video camera recorded both audio and visual information to be reviewed later.

4.2.2 Observations

We reviewed all of the annotation sessions and, for two users, we catalogued the annotation activity in detail. Figure 30 shows a visualization of the user's activities over time for their annotation session. We note five major observations:

1. Many different, but small, annotation events are highly interspersed within a short period of time. We believe that if annotation between data types required explicit actions, e.g. switching between a viewpoint control mode and gesture input mode, this intermingling would not have occurred. We note that this is consistent with Oviatt's assertion in (§2.2.2) that multimodal input "synchrony does not imply simultaneity."

   Although a variety of annotation session styles were observed, we noticed that most users segmented the annotation session into natural repeating patterns, clustering on each defect. For example, one user moved the viewpoint to locate a defect, positioned the flashlight on the area of interest, and then spoke and drew markings to comment on the defect. This series of actions was repeated for each defect the user found. Another user adopted a style that involved inspection by repeatedly orbiting around the model. When a defect was spotted, the user left orbit and zoomed in on the region of interest.

2. Users spoke the majority of the time during the annotation session, although this may have been due to slight nervousness about their performance in the task. Nevertheless, we found that their speech often occurred in parallel with viewpoint movement or gesture. Indeed, we observed that voice annotations were the predominant style of commenting in conjunction with "framing" the proper 3D view. The flashlight mode was used less frequently than the pen mode.

3. One surprising result was how users employed the snapshot feature. While the test task could be performed easily without using snapshots, we observed that some

users applied an annotation and then pulled back to make a snapshot of it. One user in particular annotated all of the defects and then made a second pass taking a snapshot of each defect with its annotation. In both of these cases, we speculate that users took snapshots to emphasize the defects and to produce a visual summary.

4. There were a few occasions where the users accidentally confused drawing on the object and drawing on a screen snapshot. Users would make an annotation, move the viewpoint and then realize that they had not been drawing on the screen snapshot, but were instead drawing on objects in the 3D scene. Users seemed to recover easily from this mistake and move quickly back to the appropriate viewpoint to either vocalize a comment, or take a snapshot and re-annotate.

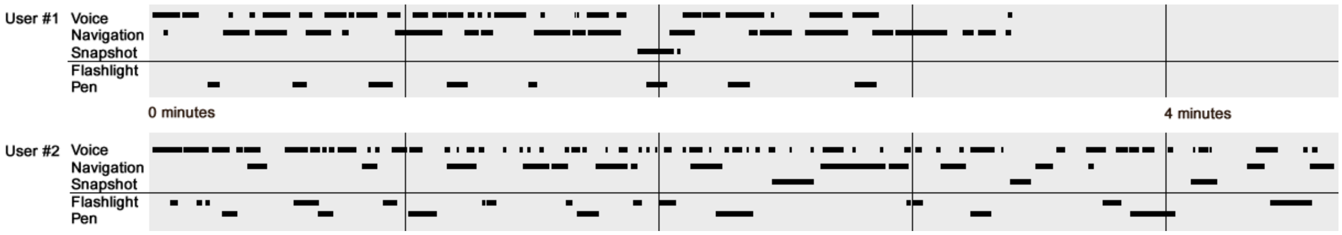


Figure 30: Annotation session for two users. Note that the multiple modes of annotation are highly interspersed.

Chapter Five:

Conclusions

In this chapter, we recapitulate our approach and give a summary of the previous chapters. We then re-present the results of our investigations and state our contributions to the field. We conclude with research directions for the future, primarily focused on leveraging spatial awareness in the annotation context.

5.1 Approach

This research took initial inspiration from the Wang Freestyle system, which offered synchronized voice and marking gestures for the annotation of image-based documents. We use this as our primary prior art and attempt to re-implement the approach, while augmenting it with 3D viewpoint information. We apply this augmented approach to the 3D domain by using the Boom Chameleon, a strong specific device that excels at chameleon-style navigation.

With this novel combination of annotation philosophy and hardware, we engaged in an exploratory survey to carve out a rich, multi-dimensional design space. We did not performing any formal user evaluations or set out to focus on any particular aspect of this space. Instead, we attempted to shed the most light on the topic with the resources available, to identify the areas of interest which warrant more rigor and formal evaluation.

The result is an annotation system which captures existing work-practice annotation styles in a minimally intrusive manner by simultaneously capturing three different modes of input. We employ a "window in hand" metaphor for the cornerstone task of navigation, resulting in a low overhead system where freedom from pre-planning and mode switching enables the user to create fluid annotations while seamlessly engaging several modes of input.

5.2 Chapter Summaries

Chapter one introduced the Boom Chameleon device and the background motivations behind our research. The viewpoint, shared view, and information capture problems (§1.1) were some of the motivations behind our research objectives of examining the characteristics of the Boom Chameleon device that make it conducive to certain types of navigation; building an annotation application which makes use of the Boom Chameleon to allow simultaneous capture of different input modalities; and evaluating the users' tendency to make use of the simultaneous data capture during annotation (§1.2).

Chapter two examined the related navigation and annotation background and research. In the navigation section (§2.1) we discussed the factors influencing the suitability of input devices to certain types of navigation. Chameleon-type navigation systems similar to the Boom Chameleon were examined, with factors that made the device easy to use. The navigation section concluded with analysis of the biases created by the Boom Chameleon. The remainder of the chapter (§2.2) reviewed annotation in 3D, modes of annotation, related systems, and laid the foundation for the annotation application.

Chapter three described the design and implementation of an annotation application. The annotation interface (§3.1), viewpoint capture, voice capture, session management and workflow were described. Techniques including pen mode, flashlight mode, snapshots, animated transitions, and titlebars were used to address the difficulties of gesture capture. Other design issues were discussed (§3.2), including the role of failure and redesign, button and snapshot placement, snapshot quality, and the size, positioning and sparsity of the model being displayed. Contexts of use were covered (§3.3) including Task, Social, and Machine Contexts. Task contexts included browsing, presentation, capture and transmission/playback. The chapter ended with system implementation (§3.4), which described both hardware and software architecture.

Chapter Four contains user evaluation and the resulting observations for the view-only Boom Chameleon (§4.1), and the setup and observations of the annotation-enabled Boom Chameleon.

5.3 Results

Analysis. Analysis of the Boom Chameleon found the use of existing motor skills, spatial perception, person-sized scale of operation, physical constraints, and Chameleon-type navigation results in specific biases that affect the navigation, view, and interaction of the device. We found that these biases facilitated low-overhead egocentric navigation that was easy to learn and use.

Iterative Design and Implementation. We built a prototype to take advantage of the ease of navigation with the Boom Chameleon device. An event-based architecture was created to support the continuous freeform capture of viewpoint, voice and gesture annotation. The interface to the resulting application was refined through informal evaluation.

Design Discussion and Rationale. Several problems arose during the annotation of models and solutions that preserve continuity and minimize intrusiveness were proposed. We employed these solutions during our implementation of viewpoint manipulation, snapshot, flashlight, and marking gesture features. Difficulties we encountered and addressed include those that concern the flow and placement of annotation markings, perspective, sparse scene annotation, and annotation layering.

System Evaluation. Informal evaluations with the Boom Chameleon found that it employs an easy to learn and operate navigation metaphor. A separate evaluation on the efficacy of the annotation facility found that participants were annotating using mixed and highly interspersed types of data. This suggests that our approach allowed users to successfully take advantage of multiple, synchronous modalities in their annotations.

5.4 Research Contributions

This thesis has described a spatially-aware display combined with the ability to capture voice, gesture and viewpoint information in a continuous, free-form fashion. We have contributed to the analysis of a unique input / output device (§2.1.5) and to the development of a novel 3D annotation application (§3.1) with associated interaction techniques. We have presented novel approaches and analysis to address problems with gesture capture, including placement of double-sided snapshots into a 3D scene which enables drawing in "empty-space" and affords$_{(N)}$ a natural layering of the annotations; animated transitions promote user orientation; and semi-transparent titlebars toggle snapshot visibility and allow the orientation and position of the hidden snapshot to be deduced (§3.1.2). In the process, we have identified and solved the problem of *ink drift* by freezing the snapshot view (§3.1.2); and introduced an alternative, real-time view during the playback of viewpoint annotations allowing for a side-by-side comparison of information from different perspectives (§3.1.3). We have defined an *intimate interface* (§3.2.2), proposed an alternate layout for an existing hierarchy (§3.3.2), and identified and suppressed the *herding effect* by using a duplicate fixed display (§4.1).

5.5 Future Work

In future work, we would like to address the many limitations of our system. For example, methods for deleting and editing portions of an annotation session may be introduced. Another avenue for exploration is the use of Chameleon navigation in combination with other types of camera controls, mappings and control/display ratios. For example, a graphical widget could be used to control the view magnification. Other issues are how similar larger-scale or smaller-scale devices change the interaction.

Other directions to explore are the contexts of use and application specific domains. For example, the Boom Chameleon could be used to produce camera shots in virtual scenes for early-production visualization in the entertainment industry. Household applications could include the use of specially instrumented spatially-aware TabletPCs to, for example, describe how well a new couch would fit into the corner of a room.

We have insights that are also applicable to other areas, such as with augmented reality applications. Since our system allows the user to keep in direct contact with others while annotating virtual objects, the resulting hand, eye, and group dynamics may well afford rich social cues while working in a shared collaborative space [3].

The work to date has focussed on capturing of annotation using the affordances$_{(G,N)}$ of the Boom Chameleon device. Some open issues surrounding the capture of the annotations concern the clutching of navigation, and new and best practices scaling and placing the model when annotating for different purposes.

A natural extension of the research into session capture is the investigation of novel interfaces for session playback. Other directions of future research include extraction of the information contained within each recording and the visualization of resulting aggregate information. Since one of the unique aspects of the Boom Chameleon is the use of the spatially-aware display for navigation, this will be the focal point for most of the future work on playback, visualization and extraction, and workflow.

Playback is often indexed by time, but one could imagine using a spatially-aware display to spatially index playback. The Boom Chameleon display could be used as a person-sized, spatial jog-wheel. Since viewpoints, gestures, and markings on a model are necessarily spatial, the recording could be indexed using the regions of 3D space surrounding the model, and the user could query specific parts of the model.

*Visualization and Extraction***.** The Salvage Station [67] investigates extraction of useful information from recordings. Similarly, rather than playing each recording in its entirety to glean information, an effective distillation of pertinent annotation information would be useful. Sitting through hundreds or thousands of annotations would be a daunting task, and at such scales even techniques such as using time-compressed playback [40] would tedious and demanding and would only reduce the time by a linear factor. Thus, the aggregation of multiple sessions of information into a single entity is another area for future research. Since the Boom Chameleon allows recorded, natural movements of the viewpoint, an example of such aggregation is to visualize the camera path densities around the model. Marketing personnel could use this to see which views of a car garner the most interest.

With the advent of new technology, there may be opportunities to obtain the spatial awareness of wireless devices at a reasonable cost [3], which in turn may allow chameleon style navigation in a portable and cheap format. Spatially-aware displays and devices will then become more commonplace. The ability to leverage these devices could bring tremendous benefits – in our case, we have shown that such awareness brings several benefits in the annotation of three-dimensional data.

5.6 Conclusion

Our work with the Boom Chameleon has demonstrated that the “window on a virtual world” metaphor is extremely quick to learn and an easy to operate 3D navigation technique. A weakness of this approach is that it allows only a limited type of navigation and requires special hardware. Nonetheless, we feel that it could be used effectively in many applications where non-technical users require the ability to inspect 3D virtual objects. Studies using our annotation system have shown that users utilized the simultaneous capture of 3D viewpoint, voice and gesture and easily transitioned between the data types. Because of this, we believe that the Boom Chameleon has the potential to be an effective tool in 3D design reviews.

References

1. **Art+Com** (http://www.artcom.de)
2. **Balakrishnan, R., Fitzmaurice, G.W. and Kurtenbach, G. (2001).** User Interfaces for Volumetric Displays**.** *IEEE Computer*, 34(2), 37-45.
3. **Billinghurst, M., and Kato, H. (2002)**. How the Virtual inspires the real: Collaborative Augmented Reality. *Communications of the ACM 45,* 7 (July 2002), 64-70.
4. **BlueSoft, Inc.** (http://www.bluesoft-inc.com)
5. **Bolt, R.A. (1980).** Put-That-There: voice and gesture at the graphics interface**.** *SIGGRAPH Computer Graphics*, 14(3), 1980, 262-270.
6. **Bourguignon, D., Cani, M-P, and Drettakis, G. (2001).** Drawing for Illustration and Annotation in 3D**.** *Proceedings of EUROGRAPHICS'01*, 114-122.
7. **Bowman, D.A. (1999).** Interaction Techniques For Common Tasks In Immersive Virtual Environments: Design, Evaluation, and Application, *Doctoral dissertation, Georgia Institute of Technology*, (http://vtopus.cs.vt.edu/~bowman/thesis/), Section 2.4, Evaluation of Immersive Virtual Environments.
8. **Brown, E., Buxton, W. & Murtagh, K. (1990).** Windows on tablets as a means of achieving virtual input devices. In D. Diaper et al. (Eds), *Human-Computer Interaction - INTERACT '90.* Amsterdam: Elsevier Science Publishers B.V. (North-Holland), 675-681.
9. **Buxton, W. (1983).** Lexical and Pragmatic Considerations of Input Structures. *Computer Graphics,* 17(1), 31-37.
10. **Buxton, W. (1995).** Integrating the Periphery and Context: A New Model of Telematics *Proceedings of Graphics Interface '95, 239-246.*
11. **Buxton, W. (1986).** Chunking and Phrasing and the Design of Human-Computer Dialogues, *Proceedings of the IFIP World Computer Congress,* Dublin, Ireland, 475-480.
12. **Buxton, W. (1986).** There's More to Interaction than Meets the Eye: Some Issues in Manual Input. In Norma, D.A. and Draper, S.W. (Eds.), (1986), *User Centered System Design: New Perspectives on Human-Computer Interaction.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 319-337.
13. **Buxton, W. (2001)**. Less is More (More or Less), in P. Denning (Ed.), *The Invisible Future: The seamless integration of technology in everyday life*. New York: McGraw Hill, 145 – 179. Also at (http://www.Billbuxton.com/LessIsMore.html)
14. **Buxton, W. & Fitzmaurice, G.W. (1998).** HMD's, Caves & Chameleon: A Human-Centric Analysis of Interaction in Virtual Space**,** *Computer Graphics*, *The SIGGRAPH Quarterly*, 32(4), 69-74.
15. **Buxton, W., Fitzmaurice, G., Balakrishnan, R., & Kurtenbach, G. (2000)**. Large Displays in Automotive Design. *IEEE Computer Graphics and Applications*, 20(4), 68-75.
16. **Buxton, W. Hill, R. & Rowley, P. (1985).** Issues and Techniques in Touch-Sensitive Tablet Input, *Computer Graphics*, 19(3), 215-224.
17. **Chalfonte, B.L, Fish, R.S. and Kraut, R.E. (1991).** Expressive Richness: A comparison of speech and text as media for revision**,** *Proceedings of the CHI'91 Conference on Human Factors in Computing Systems*, 21-26. *ACM*, 1991.
18. **Cohen, P., McGee, D., Oviatt, S., Wu, L., Clow, J., King, R., Julier, S. and Rosenblum, L,**

(1999). Multimodal Interaction for 2D and 3D Environments, *IEEE Computer Graphics and Applications*, 19(4), 10-13.

19. **Dourish, P. and Bly, S. (1992).** Portholes: supporting awareness in a distributed work group, *in CHI'92 Conference Proceedings on Human Factors in Computing Systems*, 541-547.
20. **Dourish, P., Bellotti, V. (1992).** Awareness and Coordination in Shared Workspaces, *in Proceedings of the 4th ACM Conference on CSCW*. Toronto, Canada.
21. **Elo Touch Systems** (http://www.elotouch.com/)
22. **Evers, M. (1999)**. *A Case Study on Adaptability Problems of the Separation of User Interface and Application Semantics*. Technical Report TR-CTIT 99-14. Center for Telematics and Information Technology. University of Twente, the Netherlands. Available at: http://www.ub.utwente.nl/webdocs/ctit/1/00000017.pdf
23. **Fakespace Labs** (http://www.fakespacelabs.com)
24. **Fels, S. S. (2000).** *Intimacy and Embodiment: Implications for Art and Technology,* Proceeding of the ACM Conference on Multimedia, Marina del Sol, CA, pp. 13-16, Oct., 2000.
25. **Fels, S. S. and Mase, K. (1997).** Intimacy as an index for understanding human-computer interaction. In Proceedings of the Workshop on Interactive System and Software, page p. 211, Dec 1997. (abstract at http://www.ece.ubc.ca/~ssfels/publications/WISS97.txt)
26. **Fitzmaurice, G.W. (1993).** Situated Information Spaces and Spatially Aware Palmtop Computers. *Communications of the ACM*, 36(7), 38-49.
27. **Fitzmaurice, G., Ishii, H., & Buxton, W. (1995).** Bricks: Laying the foundations for graspable user interfaces. *Proceedings of the CHI'95 Conference on Human Factors in Computing Systems*. 432-449. *ACM*, 1995.
28. **Fitzmaurice, G., Zhai, S. and Chignell, M. (1993).** Virtual reality for palmtop computers**,** *ACM Transactions on Information Systems*, 11 (3), 197-218.
29. **Foley, J.D. and Van Dam, A. (1982).** *Fundamentals of Interactive Computer Graphics.* Reading, MA, Addison Wesley.
30. **Foley, J.D., Van Dam, A., Feiner, S.K., and Hughes, J.F. (1990).** *Computer Graphics – Principles and Practice*, *2nd ed.* Addison-Wesley Publishing Co., Reading, MA, Chapter 9, 394-395.
31. **Frohlich, D. and Daly-Jones, O. (1995).** Voicefax: A shared workspace for voicemail partners**,** *Proceedings of the CHI'95 Conference on Human Factors in Computing Systems Short papers*, 308-309. *ACM*, 1995.
32. **Galyean, T.A., and Hughes, J.F. (1991).** Sculpting: An interactive volumetric modeling technique. *Computer Graphics*, Vol. 25, No. 4, pp 267-274, July 1991.
33. **George, R. Varkki, and Rajiv Ramanathan. (1999)**. Combining Egocentric and Exocentric Views: Enhancing the Virtual Worlds Interface. *6th International Conference on Computers in Urban Planning and Urban Management*. Venice, Italy.
34. **Gibson, J.J. (1979).** The Ecological Approach to Visual Perception. Boston: Houghton Mifflin.
35. **Grossman, T., Balakrishnan, R., Kurtenbach, G., Fitzmaurice, G.W., Khan, A., Buxton. W. (2001).** Interaction techniques for 3D modeling on large displays. *Proceedings of ACM I3D 2001 Symposium on Interactive 3D Graphics*, p. 17-23.
36. **Guimbretire, F. Stone, M. and Winograd, T. (2001).** Fluid Interaction with High-resolution Wall-size Displays, *in Proceedings of UIST 2001*, 21-30, ACM Press.

37. **Hand, C. (1997).** A Survey of 3D Interaction Techniques. *Computer Graphics Forum*, 16(5): 269-281.

38. **Hardock, G., Kurtenbach, G., and Buxton, W.A. (1993).** Mate: A marking based interface for collaborative writing**.** *Proceedings of the UIST'93 Symposium on User Interface Software & Technology*, 259-266. *ACM,* 1993.

39. **Harmon, R., Patterson, W., Ribarsky, W. and Bolter, J. (1996).** The Virtual Annotation System**,** *Proceedings of Virtual Reality Annual International Symposium (VRAIS'96)*, 239-245.

40. **He, L. and Gupta, A. (2001).** Exploring benefits of non-linear time compression. *ACM Multimedia 2001*. 382-391

41. **Hinckley, K. (1997).** Haptic Issues for Virtual Manipulation. *Ph.D. dissertation, University of Virginia, Dept. of Computer Science.* Chapter 4, Design Issues in Spatial Input, 81-96.

42. **Hinckley, K, Pausch, R, Goble, JC, and Kassell, NF. (1994).** Design Hints for Spatial Input, *Proceedings of the UIST'94 Symposium on User Interface Software & Technology,* 213-222. *ACM,* 1994.

43. **Igarashi, T., Hughes J.F. (2001):** Voice as sound: using non-verbal voice input for interactive control. *Proceedings of the UIST 2001 Symposium on User Interface Software & Technology.* 155-156 *ACM.* 2001.

44. **Iris Annotator** (SGI: http://www.sgi.com/software/annotator)

45. **Jacob, R., Sibert, L., McFarlane, D., & Mullen, M. (1994).** Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction*, 1(1). p. 3-26.

46. **Jung T., Gross M. D. and Do E. Y., (2002)**, Annotation and Sketching on 3D Web Models, In *Proceedings of the IUI'02 Conference on Intelligent User Interfaces*, 95-102. *ACM,* 2002.

47. **Klemmer, S.R., M.W. Newman, and R. Sapien**. **(2000).** The Designer's Outpost: A Task-Centered Tangible Interface for Web Site Information Design. *In Proceedings of CHI, Human factors in computing systems 2000*: ACM Press. pp. 333-334 2000.

48. **Kuzuoka, H., Ishimoda, G., Nishimura, Y., Suzuki R., and Kondo K. (1995)**
(http://www.kuzuoka-lab.esys.tsukuba.ac.jp/lab/gesturecam/bear.mpg**)**
Can the gesturecam be a surrogate?, *in ECSCW'95*

49. **LaViola, J., Holden, A., Forsberg, D. and Bhuphaibool, D. (1998).** Collaborative Conceptual Modeling Using the SKETCH Framework, *Proceedings of the First IASTED International Conference on Computer Graphics and Imaging*, 154-158.

50. **Levine, S. and Ehrlich, S. (1991).** The Freestyle System: A Design Perspective. In Klinger, A. (Ed.) *Human-Machine Interactive Systems, Plenum 3-21*. Also in Baecker, R.M., Grudin, J., Buxton, W., and Greenberg, S. (Eds.) *Readings in Human-Computer Interaction: Toward the Year 2000*, 2nd edition. 871-880.

51. **Lienhart, Rainer. (2000).** A System for Effortless Content Annotation to Unfold the Semantics in Videos. In *Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'00)*, 45-50.

52. **Lindeman, R., J. Sibert, and J. Hahn. (1999).** Hand-Held Windows: Towards Effective 2D Interaction in Immersive Virtual Environments. *IEEE Virtual Reality*, 1999.

53. **Loughlin, M. and Hughes, J. (1994).** An Annotation System for 3D Fluid Flow Visualization. In *Proc. IEEE Visualization '94*, 273-279.

54. **Maeda, J., Greenwold, S.** Installation. *Aesthetics and Computation Group, MIT.* (http://acg.media.mit.edu/people/simong/installationNew/cover.html)

55. **Marshall, C.C. (1997).** Annotation: from paper books to the digital library. *Proceedings of the Second ACM International Conference on Digital Libraries*, 131 – 140.

56. **McGrenere, J.**, (2002). The design and evaluation of multiple interfaces: A solution for complex software. Dissertation. Department of Computer Science, University of Toronto, Toronto, ON. Canada. Section 2.3, What is the User's Experience of Complex Software?, 13-21.

57. **McGrenere, J.**, **and Ho, W**. (2000). Affordances: Clarifying and evolving a concept. *Proceedings of Graphics Interface 2000.*

58. **Moore, F. R. (1988).** The dysfunctions of midi. Computer Music Journal, 12(1):19–28, Spring 1988. Cited in Fels, S., *Intimacy and Embodiment: Implications for Art and Technology,* Proceeding of the ACM Conference on Multimedia, Marina del Sol, CA, pp. 13-16, Oct., 2000.

59. **Moran, T. P., Chiu, P., Harrison, S., Kurtenbach, G., Minneman, S., and van Melle, W. (1996).** Evolutionary engagement in an ongoing collaborative work process: a case study. *Proceedings of CSCW '96*. ACM, 150-159.

60. **Moran, T., Palen, L., Harrison, S., Chiu, P., Kimber, D., Minneman, S., van Melle, W., and Zellweger, P. (1997).** I'll get that off the audio: salvaging in a multimedia meeting**.** *Proceedings of the CHI'97 Conference on Human Factors in Computing Systems Interaction* 202-209. *ACM*, 1997.

61. **Morris, J.H., Neuwirth, C.M. et al.** Virtual Work Room. *School of Computer Science, Carnegie Mellon University.* (http://muffin.hss.cmu.edu/vwr/web/overview.html)

62. **Norman, D.A. (1988).** The Psychology of Everyday Things. New York: Basic Books.

63. **O'Hara, K., Sellen, A., (1997).** A comparison of reading paper and on-line documents. *Proceedings of the CHI'97 Conference on Human Factors in Computing Systems Interaction*, 335-342. *ACM,* 1997.

64. **Oviatt, S. L. (1999).** Ten myths of multimodal interaction, *Communications of the ACM*, Vol. 42, No. 11, November, 1999, pp. 74-81.

65. **Pedersen, E.R., McCall, K., Moran, T.P., Halasz, F.G., (1993).** Tivoli: an electronic whiteboard for informal workgroup meetings. In *Proceedings of CHI'93 Conference on Human Factors in Computing Systems Interaction*. 391-398. *ACM,* 1993.

66. **Poupyrev, I., Tomokazu, N., and Weghorst, S. (1998).** Virtual Notepad: Handwriting in Immersive VR, In *Proceedings of IEEE Virtual Reality Annual International Symposium'98 (VRAIS'98)*, 126-132.

67. **Razzaque, S., Kohn, Z., and Whitton, M. (2001).** Redirected Walking, *Eurographics 2001* Short Presentation

68. **Rekimoto, J. and Nagao, K. (1995).** The World through the Computer: Computer Augmented Interaction with Real World Environments, *Proceedings of the UIST'95 Symposium on User Interface Software & Technology*, 29-36. *ACM,* 1995.

69. **Russo dos Santos, C., Gros, P., Abel, P., Loisel, D., Trichaud, N., and Paris, J.P. (2000)**: Metaphor-Aware 3D Navigation. *INFOVIS 2000*. 155-166

70. **Salzman, M., Dede, C., Loftin, B., and Ash, K., (1998),** VR's frames of reference: A visualization technique for mastering abstract information spaces, *Proceedings of 3rd International Conference on Learning Sciences*, 249-255.

71. **Salzman, M., Dede, C., Loftin, B. (1999),** VR's Frames of Reference: A Visualization Technique for Mastering Abstract Multidimensional Information, *Proceeding of the CHI'99 Conference on Human Factors in Computing Systems*. 489-495. *ACM***,** 1999.
72. **Schmalstieg, D., Encarnacao, L.M. and Szalava, Z. (1999).** Using transparent props for interaction with the virtual table," In *Proceedings SIGGRAPH'99 Symposium on Interactive 3D Graphics*, 147-154. *ACM,* 1999.
73. **Small, D. and Ishii H. (1997).** Design of Spatially Aware Graspable Display*, CHI'97 Conference on Human Factors in Computing Systems Electronic Publications*. *ACM*, 1997**.**
74. **SpaceMouse** (http://www.3dconnexion.com/articles/magelus.html**)**
75. **Stellingwerff, Martijn. (1999).** SketchBoX. *Proceedings of Architectural Computing from Turing to 2000. eCAADe*, ISBN 0-9523687-5-7, Liverpool (UK) 15-17 September 1999, 491-497.
76. **Stoakley, R., Conway, M. and Pausch, R. (1995).** Virtual reality on a WIM: Interactive worlds in miniature. *Proceedings of the CHI'95 Conference on Human Factors in Computing Systems Interaction*, 265-272. *ACM*, 1995
77. **Weber, K., and Poon, A. (1994).** Marquee: A Tool for Real-Time Video Logging, *Proceedings of the CHI '94 Conference on Human Factors in Computing Systems Interaction*, 58-64. *ACM*, 1994.
78. **Whittaker, S. Davis, R. Hirschber, J., and Muller, U. (2000).** Jotmail: a voicemail interface that enables you to see what was said, *Proceedings of the CHI'2000 Conference on Human Factors in Computing Systems Interaction*, CHI Letters 2(1), 89-96. *ACM*, 2000.
79. **Whittaker, S., Hyland, P., and Wiley, M. (1994).** Filochat: Handwritten notes provide access to recorded conversations. *Proceedings of the CHI'94 Conference on Human Factors in Computing Systems Interaction*, 271-277. *ACM*, 1994.
80. **Wilcox, L.D., Schilit, B.N., and Sawhney, N. (1997).** Dynomite: A Dynamically organized Ink and Audio Notebook. *Proceedings of CHI'97*, 186-193.
81. **Ware, C. (2000).** Information Visualization: Perception for Design. San Mateo, CA: *Morgan Kaufman publishers,* 2000.
82. **Ware, C. and Osborne, S. (1990).** Explorations and Virtual Camera Control in Virtual Three Dimensional Environments. *Computer Graphics.* 24(2) 175-183.
83. **WindowsVR** (http://www.abs-tech.com/Produtos/3D_VR/VR-Hardware/hmds1/Virtual_Research/win_vr.html)
84. **Zhai, S. (1995).** Human Performance in Six Degree of Freedom Input Control**.** *Doctoral Dissertation, University of Toronto.* Chapter 2, The Effects of Device Resistance and Transfer Function in 6 DOF Manipulation.
85. **Zhai, S., & Milgram, P. (1993).** Human Performance Evaluation of Manipulation Schemes in Virtual Environments. Proceedings of IEEE VRAIS. 155-161.

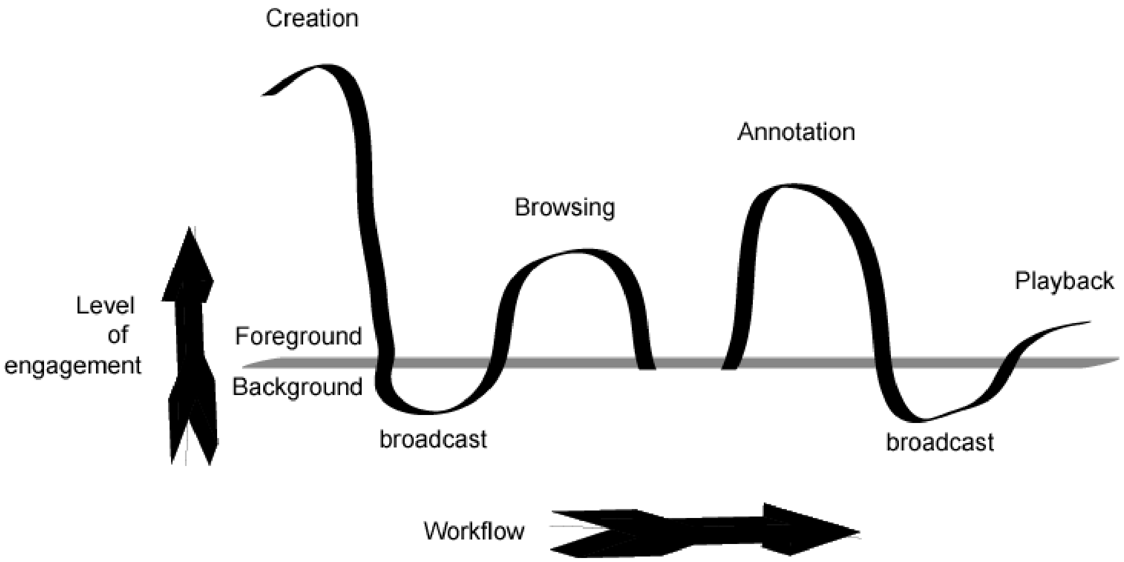
Appendix A: Egocentric and Exocentric Viewpoints

Not to be confused with egocentric and exocentric control metaphors, viewpoints may also be classified as egocentric or exocentric. This means that the viewpoint from the first-person or third person perspective. The first-person is from the perspective of a phenomenon we are interested in, for example looking *from* the centre of an apple outwards; conversely, the third-person perspective is centred on a phenomenon external to the viewpoint, for example from the *outside looking in* towards the apple. In the published literature, these are respectively termed *egocentric* and *exocentric viewpoints*.

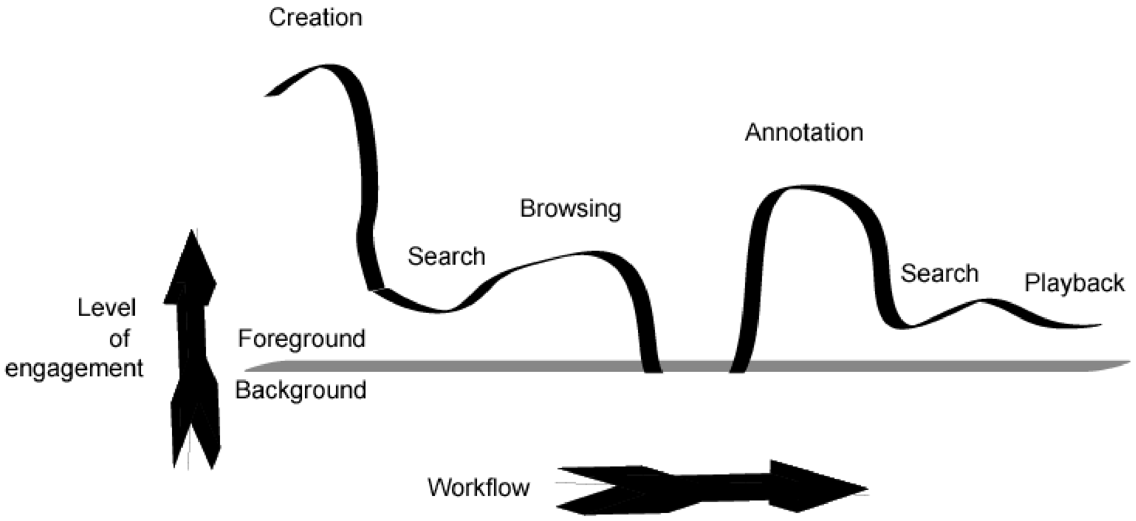
Indeed, George and Ramanathan characterize the egocentric view as the viewpoint portrayed from the first-person perspective [33]. The authors note that this contributes to a make the experience "seem real". However, although the immersiveness of the egocentric view may help ground the user with the situation at hand, this may also be disorienting in unfamiliar environments. Using the example of a traveller walking through a forest, one could imagine trying to walk in a straight line through the trees, but becoming disoriented and walking in a crooked line. In contrast, if one were to enjoy a bird's-eye view of the traveler from overhead, it would be very easy to determine the straight path. Such an overhead view or any other third-person perspective is known as an *exocentric view*. An exocentric view is more conducive to the user being aware of surrounding information and thus staying oriented. It is characterized with the viewpoint as being removed from the phenomena of interest, an overhead map of the forest being one example. Compared to an egocentric view, the exocentric view employs a distanced vantage point, and may contain much different information about the world, such as the overall shape of the forest, the direction to the nearest river, or other information that may not be easily accessible from the egocentric perspective. Exocentric viewpoints are acknowledged to provide a significant contribution to contextual understanding of the environment and of specific concepts relating to it [33, 70, 71]. The Boom Chameleon device is currently most conducive to object inspection using an exocentric view.

Appendix B: Interaction of the Task and Social Contexts

A model described in [10] supports "holistic thinking" about the "symbiotic relationships among a suite of applications." Part of this model describes tasks in the *foreground*, in the "fore of human consciousness" The author argues that although foreground activities, such as those described in §3.3.1 maybe very rich, they are missing what is available in simple shared physical space, a "peripheral awareness" of phenomena. *Background* tasks, on the other hand, are "tasks that take place in the periphery" with non-engaged, relatively longer-term connections. By having the background tasks supporting "peripheral awareness" with appropriate transitions to the foreground, the author suggests we can remove unnecessary complexity from a task



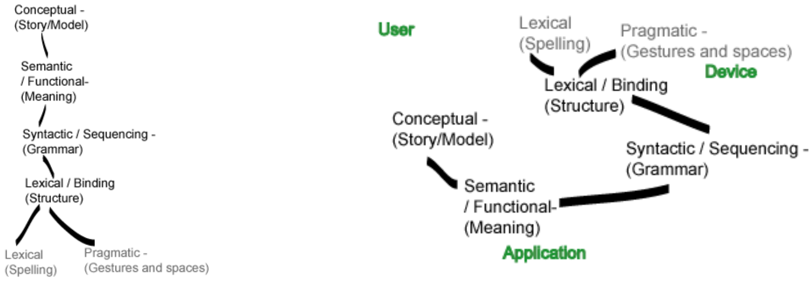
After models are created and new annotations are composed with our system, they automatically appear in the graphical browser displayed on the Boom Chameleon (§3.1.3). This 'broadcasting' puts the work into the shared periphery, and helps support the "background social ecology" [10] of the workflow. The level of engagement changes from the foreground activity of creation into the background of transmission and broadcast / display. When users choose to browse, annotate or playback annotation sessions, the workflow moves again into a foreground activity.



This contrasts to the scenario where the user must explicitly search for models or annotations of interest. With increased foreground task requirements, the complexity of the overall task is increased. Thus by acting as an awareness server, the Boom Chameleon system contributes to the social context of the material and makes it easier for the user to perform his or her task.

Appendix C: Interface Dependence on the User, Application, and Device

The taxonomy described by Foley et al. in is a simple hierarchy (Figure 5 and Figure 27a, reprinted below) that suggests a separable layering of concerns. Yet the conceptual and semantic layers are often highly dependent on the user and the application: the user's model and understanding of what can be expressed is shaped by the user's experience with the application. As Buxton points out [9], the pragmatics of the device also strongly effect the interaction. This suggests organizing the layout of the hierarchy to highlight each layer's dependence on the user, application and device (see Figure 27b, reprinted below). We can use this diagram to help analyse impact on the hierarchy when we disturb any part of the user, application, or device.


Conceptual - (Story/Model)
Semantic / Functional- (Meaning)
Syntactic / Sequencing - (Grammar)
Lexical / Binding (Structure)
Lexical (Spelling)
Pragmatic - (Gestures and spaces)
User
Lexical (Spelling)
Pragmatic - (Gestures and spaces)
Device
Lexical / Binding (Structure)
Conceptual - (Story/Model)
Syntactic / Sequencing - (Grammar)
Semantic / Functional- (Meaning)
Application

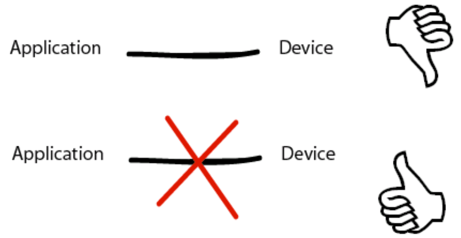
An explanation of the layout follows.

Pragmatic component: This component has the most dependence on the device, but is also partially dependent on the user, who performs the actions on the device. For each user, some gestures and actions may be easier than others. *Lexical component*: The number of key presses or degrees rotated is dependent on both the ability of the user and the affordances$_{(G)}$ of the device. *Syntax*: The series of key presses or movements constitute a meaningful act is in the domain between the application, and the device. *Semantic*: The application has primary responsibility for determining the meaning of each cogent act, which is expressed through the syntactic grammar. However, if the act is to 'deposit ink on a surface' the interpretation of the semantic meaning is up to the user. *Conceptual*: Different users may have different views on the functionality of an application, even on the same device. For some people, a web browser is something used to read the news. For others, it is a conduit for electronic auctions, or a picture browser.

Appendix D: The Effect of Application and Device Generalization on Usability

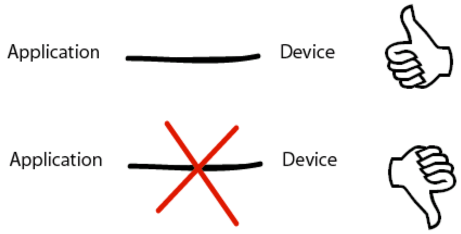
Device independence, application independence and the separation of semantics. Application independence, or being able to perform different tasks on one machine, is similar in many ways to the seemly opposite problem of device independence. We will therefore examine the relative symmetry of the problems: The idea of device independence is to allow device behaviour to be abstracted and applications to use many different, but logically equivalent devices. Software architectures often assume device independence through a clean separation of the interface and application semantics. In fact, the lack of separation is perceived as an application adaptability problem [22]. This suggests that the issue of application independence is similar. By weakening the link between interface and application, we can create application independence.

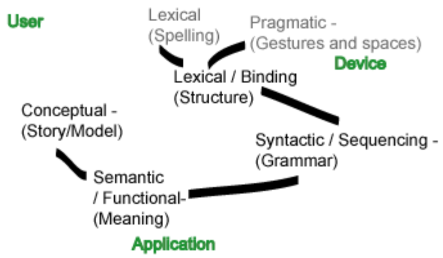
Device and Application Independence



Although this facilitates experimentation, the ability to logically change devices does not guarantee a similar degree of usability for each device. In fact, by separating the layers we might destroy the close fit between device and application that promotes ease of use (§2.1.2.1).

Hierarchy alignment and Usability





Device independence results in weakened usability. Buxton supports this suggestion in [12], claiming that technical interchangability does not extend to usability, due to the idiosyncratic differences. The layout in Figure 27b (reprinted above) may help illustrate how idiosyncratic differences effect the interaction. Decoupling the device and application means weakening the link between the semantic and syntax layers. Replacing the device changes the hierarchy elements in the device area of the diagram, with direct repercussions for most of the hierarchy. The underlying model suggests that, in a system with an existing strong device-application coupling, changing the devices will alter the lexical and pragmatic components of interaction. This in turn affects the alignment of the entire hierarchy.

Generalization increases task complexity. Whether an application is altered to accommodate many devices, or a single device is used for many different applications, the act of generalizing either will most likely weaken a coupling between application and device semantics and grammar, resulting in increased task complexity. The author of [12] agrees, at least in part, with the suggestion. He claims that if a general-purpose input device is customized for a specific application, it will “come at the expense of increased cost in the complexity of the control structure[11].”

[11] A simple example would be a dedicated hammer, versus a Swiss Army Knife with a hammer, and 40 other tools. There is an increased complexity that accompanies the increase in generality.