A Web-Based Usability Tool for Assessment of Small-Screen GUIs

by

Theodore L. Weatherly, III

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Computer Science and Engineering

at the Massachusetts Institute of Technology

August 28, 2000

Author__________________
Department of Electrical Engineering and Computer Science
August 28, 2000

Certified by__ 8/28/2000
Dr. Nishikant Sonwalker
Director, Hypermedia Teaching Facility
Thesis Supervisor

Accepted by______________________________
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

A Web-Based Usability Tool for Assessment of Small-Screen GUIs

by
Theodore L. Weatherly, III

Submitted to the
Department of Electrical Engineering and Computer Science

August 28, 2000

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Computer Science and Engineering

ABSTRACT

In this thesis, we present a framework for designing and analyzing web-based, hyperactive usability environments. We present preliminary requirements to assess the utility of hyperactive environments in the development of graphical user interfaces for small-screen digital devices. We also discuss design techniques for the creation of these hyperactive environments, as well as analytical evaluations for scrutinizing the reliability and worthiness of such environments in a usability-oriented setting.

The data collected from a small group of test participants indicates that:

1. The hypermedia usability tool provides ample performance and attitude assessment information from which to derive worthwhile usability conclusions.
2. The web-enabled mechanism for interacting with the GUI prototypes is not only reliable but also very effective.
3. Tab navigation within small-screen digital device GUIs is highly effective and learnable, and is most preferred by users.
4. Bottom-aligned navigation within small-screen digital device GUIs, though not the most effective or learnable arrangement, is preferred highest.

Thesis Supervisor: Nishikant Sonwalker
Title: Director, Hypermedia Teaching Facility

Acknowledgements

I would first like to thank God for blessing me with an intelligent and creative mind to enable me to even be here at MIT. His unconditional support has truly carried me through these past five fast-paced years, and his guidance has steered me toward a Master's thesis that I truly enjoyed completing.

Special thanks goes to my thesis advisor, Dr. Nishikant Sonwalker, for his generous supervision of my research over the past year. I would especially also like to thank Dr. Amar Gupta for connecting me with both Dr. Sonwalker for guidance on my thesis and SiPix Imaging for financial assistance of my research.

I certainly cannot neglect the motivational support of my loving family. Dad, thank you for your words of wisdom to help me step back from all the tests, problem sets, and papers and gain perspective on my academic endeavors here at MIT, both during my undergraduate and graduate years. Mom, I cannot even express in words how deeply I appreciate all the counseling you've given me over my college years to help pull me through my darkest hours of confusion and doubt. Kelli, thanks for your silly phone calls to help cheer me up when I was down. And Grandma, I'm grateful for the pride you've shown in me for the many sacrifices I've made in striving to achieve my Bachelor's and Master's degrees.

I'd like to give "shout outs" to all my close friends here at school for their humor, sympathy, academic help, and emotional support. The list goes as follows: Andrew "You're still a freak!" Maywah, Damon "Playa" Lewis, Damian "(Big) D" Isla, and Jeremy "Baby J" Heidel. Special dap goes out to my fellow "crew" members: Dwaine "I'm in the shithouse, dude" Clarke, Andres "Aww yeah!" Tellez, Ron "Let's push through this problem" Demon, Daisy "Verification Test" Paul, and Mayli "Artistic" Khoe.

Finally, I'd like to thank all experiment users for their participation and usability input, which of course was critical to the success of this thesis.

:)-

Table of Contents

Acknowledgements **3**

Table of Contents **5**

List of Tables **9**

List of Figures **11**

Chapter 1 Introduction **13**

Chapter 2 Design of Experiment Prototype GUIs **19**

2.1 Introduction 19

2.2 Variable GUI Elements of Small-Screen Digital Displays 19

2.3 Process for Selecting GUI Variable Elements 21

2.4 Process for Selecting Virtual Device's Mechanical Frame 22

2.5 Detailed Description of Navigation Scheme GUIs 24

2.5.1 Sliding-Icon Navigation Scheme GUI 24

2.5.2 Button-Menu Navigation Scheme GUI 25

2.5.3 Tab Navigation Scheme GUI 26

2.6 Detailed Description of Navigation Alignment GUIs 26

2.6.1 Bottom-Aligned Navigation Alignment GUI 27

2.6.2 Left-Aligned Navigation Alignment GUI 27

2.6.3 Top-Aligned Navigation Alignment GUI 28

2.6.4 Right-Aligned Navigation Alignment GUI 29

Chapter 3 Design of Web-Enabled Usability Tool **30**

3.1 Introduction 30

3.2 Usability Testing – What Are We Measuring? 31

3.2.1 Measurable Characteristics that Define of Usability 31

3.2.1.1 Effectiveness as a Measurable Usability Characteristic 31

3.2.1.2 Learnability as a Usability Characteristic 32

3.2.1.3 Flexibility as a Usability Characteristic 32

3.2.1.4 Attitude as a Usability Characteristic 32

3.2.2 Usability Characteristics Appropriate for SSDDs 33

3.3 Usability Evaluation Techniques 34

3.4 Available Hypermedia Technologies 37

3.5 Technologies Appropriate For Survey and Experimental-Based Usability Testing38

3.6 Use of Technologies in Tool Creation 39

3.6.1 User Profile Survey Page 40

3.6.2 Prototype Task-Testing Page 40

3.6.3 Prototype Evaluation Page 41

3.6.4 Overall Impressions Page 41

3.6.5 Results Summary and Submission Page 42

Chapter 4 Application of Statistical Analyses to the Hyperactive Usability Environment 43

4.1 Introduction 43

4.2 Data Cleaning 43

4.3 Characterization of the End User 46

4.4 Performance Measures of the Experiment GUIs 47

4.4.1 Grade-Based Overall Performance Evaluation 47
4.4.2 Comparative Performance Evaluation 49
4.4.2.1 Completion-Time Performance Evaluation for the Navigation Schemes. 50
4.4.2.2 Button-Press Performance Evaluation for the Navigation Schemes 51
4.4.2.3 Completion-Time Performance Evaluation for the Navigation Alignments 53
4.4.2.4 Button-Press Performance Evaluation for the Navigation Alignments 55
4.5 Attitude Measures of the Experiment GUIs 57
4.5.1 Question-Based Attitude Assessment for Navigation Schemes 58
4.5.1.1 Evaluation of Task-Specific Questioning 58
4.5.1.2 Evaluation of Prototype-Specific Questioning 59
4.5.2 Rank-Based Attitude Assessment for Navigation Schemes 61
4.5.3 Question-Based Attitude Assessment for Navigation Alignments 63
4.5.4 Rank-Based Attitude Assessment for Navigation Alignments 66
Chapter 5 Conclusions and Further Work 69
Appendix A: GUI Prototyping Tables and Figures 74
A.2 Navigation Scheme GUI Diagrams 75
A.3 Navigation Alignment GUI Diagrams 78
Appendix B: Screenshots of the SSDD Usability Tool 82
Appendix C: Question-Based Attitude Assessment Graphs and Tables 90
C.1 Navigation Scheme Attitude Assessments 90
C.2 Navigation Alignment Attitude Assessments 96
References 98

Bibliography **99**

List of Tables

Table 3-1 Overview of the Five Evaluation Methods for Gauging Usability 35

Table 3-2 Summary of Available Hypermedia Technologies 38

Table 4-1 Statistical Data of Task-Completion Times for the Navigation Scheme Tasks 51

Table 4-2 Statistical Data of Grades for the Number of Task-Completion Buttons Pressed During Navigation Scheme Task Testing 52

Table 4-3 Statistical Data of Task-Completion Times for the Navigation Alignment Tasks 54

Table 4-4 Statistical Data of Grades for the Number of Task-Completion Buttons Pressed During Navigation Alignment Task Testing 56

Table 4-5 Navigation Scheme Rankings 62

Table 4-6 User Comments Validating the Influence of Mechanical Design on Alignment Attitude 65

Table 4-7 Navigation Alignment Rankings 67

Table A.1-1 Test Matrix of Variable GUI Elements 74

Table A.2-1 Diagrams Illustrating the Sliding-Icon GUI Scheme 75

Table A.2-2 Diagrams Illustrating the Button-Menu GUI Scheme 76

Table A.2-3 Diagrams Illustrating the Tab GUI Scheme 77

Table A.3-1 Diagrams Illustrating the Bottom-Aligned GUI Scheme 78

Table A.3-2 Diagrams Illustrating the Left-Aligned GUI Scheme 79

Table A.3-3 Diagrams Illustrating the Bottom-Aligned GUI Scheme 80

Table A.3-4 Diagrams Illustrating the Left-Aligned GUI Scheme 81

Table C.1-1 Assessment Data for "Does the sequence of activities required to complete this past task follow what you would expect?" 90

Table C.1-2 Assessment Data for "Is it clear what stage the camera has reached during your navigation for this past task" 91

Table C.1-3 Assessment Data for "Does information appear to be organized logically on the screen?" 92

Table C.1-4 Assessment Data for "Are the navigation menus clearly separated from the image background?" 93

Table C.1-5 Assessment Data for "Is the method of selecting options consistent throughout the system?" 94

Table C.1-6 Assessment Data for "Does navigation using this navigation style feel natural?" 95

Table C.2-1 Assessment Data for "Are the menu options aligned properly on the screen?" 96

Table C.2-2 Assessment Data for "Does the alignment of the navigation menus feel natural?" 97

List of Figures

Figure 1-1 Iterative GUI Design Process 15

Figure 2-1 Mechanical Design Chosen for the SSDD GUI Experiment 23

Figure 4-1 Histograms of User Grades with Gaussian Distribution 48

Figure 4-2 Graph of Comparative Task-Completion Times for the Navigation Scheme Tasks 50

Figure 4-3 Graph of Comparative Grades for the Number of Task-Completion Buttons Pressed During Navigation Scheme Task Testing 52

Figure 4-4 Graph of Comparative Task-Completion Times for the Navigation Alignment Tasks 54

Figure 4-5 Graph of Comparative Grades for the Number of Task-Completion Buttons Pressed During Navigation Alignment Task Testing 55

Figure 4-6 Summative Navigation Scheme Rankings 63

Figure 4-7 Summative Navigation Scheme Rankings 68

Figure B-1 Experiment Introduction Page 82

Figure B-2 User Profile Survey Page 83

Figure B-3 Alert Box Shown Upon Entrance To the Initial Task-Test Page 83

Figure B-4 The Shockwave Prototype Downloading on the Task-Testing Page 84

Figure B-5 The Shockwave Prototype Shown After Being Downloaded 85

Figure B-6 Shot of Screen During Testing of the First Task for the Sliding-Icon Scheme 86

Figure B-7 Prototype Evaluation Page for the Sliding-Icon scheme 87

Figure B-8 Overall Impressions Page 88

Figure B-9 Results Summary and Submission Page 89

Figure C.1-1 Graphical Display of Assessments for "Does the sequence of activities required to complete this past task follow what you would expect?" 90

Figure C.1-2 Graphical Display of Assessments for "Is it clear what stage the camera has reached during your navigation for this past task?" 91

Figure C.1-3 Graphical Display of Assessments for "Does information appear to be organized logically on the screen?" 92

Figure C.1-4 Graphical Display of Assessments for "Are the navigation menus clearly separated from the image background?" 93

Figure C.1-5 Graphical Display of Assessments for "Is the method of selecting options consistent throughout the system?" 94

Figure C.1-6 Graphical Display of Assessments for "Does navigation using this navigation style feel natural?" 95

Figure C.2-1 Graphical Display of Assessments for "Are the menu options aligned properly on the screen?" 96

Figure C.2-2 Graphical Display of Assessments for "Does the alignment of the navigation menus feel natural?" 97

Chapter 1

Introduction

A growing trend in today's culture is toward the use of portable digital devices. Ever since their creation, PDAs, cell phones, and digital cameras have gained widespread use in helping people organize their information, communicate with others, and capture pictures to share them electronically with others. These devices are becoming so popular, in fact, that their functionality is actually expanding to meet the demands of consumers. Cell phones now exist, for example, that allow users to surf the World Wide Web (WWW) and check their email. Certainly, one can imagine that as time progresses and society becomes more technology-driven, the demand for hand-held devices will increase even greater. In the process, functionality available on these devices will further expand. Critical to the success of these devices, however, is the development of small-screen graphical user interfaces that can best provide a user quick, intuitive access to this functionality.

This implication is easily realized when considering the development of the personal computer. Much like many of today's hand-held devices, the graphical user interface (GUI) for the personal computer started as a mere textural display. This interface did enable users some basic features, such as word processing and programming. As time progressed, though, computer users demanded greater functionality and improved data display. The introduction of Apple's MacOS® and Microsoft's Windows® GUI's over the past decade has, in many ways, fulfilled these demands. As a result, the personal computer has become a part of approximately 35 million U.S. households today. [1]

The GUI for portable digital devices, in many respects, presents a new field of study that must now be explored. The concern that arises, then, is indeed *how* this area should be explored. The traditional approach to iterative GUI design consists of six stages, as shown in Figure 1-1. [2] In the first two stages, an interactive interface prototype is designed and created, based on several factors, such as the product's desired functionality and characteristics of target end-users. Next, this prototype is distributed to representative users of the product and tested, whereupon GUI usability information is extracted. In turn, the gathered information is analyzed, suggesting areas for improvement. These improvements are then incorporated into a redesign of the interface, and usability testing is conducted again with a new interface prototype. This process repeats until the usability objectives for the prototype are sufficiently met. Eventually, this prototype is converted to appropriate code form for the device and field-tested.

In most instances, the prototype used throughout the GUI design, test, and redesign process is deployed on its corresponding physical device. The reason for doing this is obvious: when *designing* a GUI prototype for a personal computer application, a usability engineer would naturally want the prototype *tested* on a personal computer to ensure that

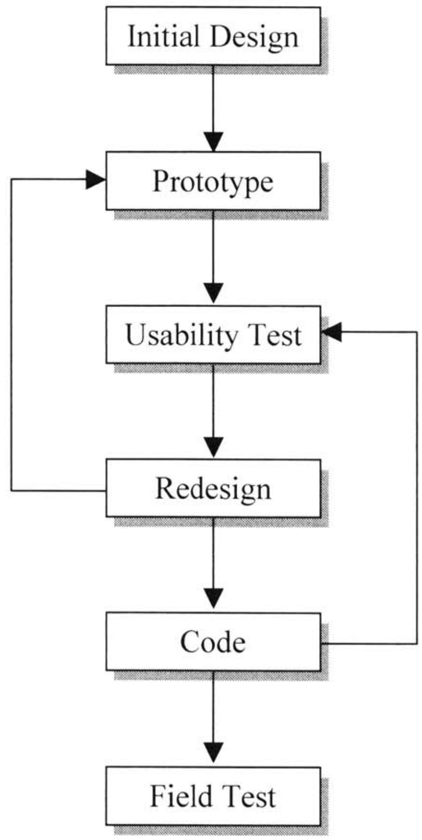


Figure 1-1 Iterative GUI Design Process

the collected usability results are reliable. In the context of GUI design for small-screen digital devices (SSDDs), though, there are several drawbacks to usability testing with such a genuine device environment. Unlike personal computers, most embedded devices are not adapted to accommodating added software. As such, in order to collect usability data, developers would need to send users both the software and hardware necessary for testing sample prototypes, which can get very expensive, especially if the test devices are not returned. Another option is assembling users into a usability lab to perform testing. This testing scheme is perhaps more affordable, but difficulties still remain. GUI developers, for instance, often have limited direct access to representative users for usability testing. And even when there are enough users to sample from, scarce lab resources may pose problems coordinating test subject participation.

Under these considerations, usability testing would be likely conducted better using an Internet-based environment. The results of case studies from research at Eastman Kodak, in fact, show that remote usability evaluation over the Internet using video teleconferencing has great potential as an extension to usability lab testing. [3]

Usability testing in a simulated Web-based environment is a similar Internet-based alternative that has yet to be explored. Testing usability over the World Wide Web provides numerous advantageous over a traditional lab setting:

- Usability testing over the WWW can save the usability engineer both time and money: He can create one usability experiment to be used by all test subjects.
- Using the WWW makes testing convenient for the user, thereby improving test response: A person can participate in the experiment in his own natural environment (i.e. home, work), and at a time that is most convenient for him.
- The WWW spans nearly the entire globe, allowing the usability engineer to reach audiences that were perhaps once too distant.
- Usability testing over the WWW can expedite the iterative design process: Results gathered from the experiments can be automatically coded and stored into a database for analysis, enabling rapid prototyping and early evaluation of the interfaces being tested.

- Results from usability tests can be easily published on the WWW for both the usability engineer and the participant to see.

Despite these benefits, the feasibility of testing in such a simulated environment still remains in doubt. Usability engineers must question several aspects of usability testing in a Web-based environment:

- Reliability: To what extent do results from the usability tests reflect results gathered using real devices?
- Content: Is there a sufficient amount of reliable information that can be gathered?
- Worthiness: Upon analysis of the trusted test results, what conclusions can be drawn, and how valuable are these conclusions toward improving the prototype's design?

Research at the MIT Hypermedia Teaching Facility provides support for the use of the WWW as a viable medium for educational learning. In a recent study at the Hypermedia Lab, Kashambuzi found that students could learn course material over the WWW at rates equivalent to being in the classroom. [4] The feasibility of usability testing over the WWW, though, is concerned with the opposite scenario: how well can usability engineers learn from users?

In this thesis, we present a framework for designing and analyzing web-based, hyperactive usability environments. We present preliminary requirements to assess the utility of hyperactive environments in the development of graphical user interfaces for small-screen digital devices. We also discuss design techniques for the creation of these hyperactive environments, as well as analytical tests and evaluations for scrutinizing the reliability and worthiness of such environments in a usability-oriented setting.

The data collected from a small group of test participants for the current design indicates that:

1. The hypermedia usability tool provides a wealth of relevant, reliable performance and attitude assessment information from which to derive worthwhile usability conclusions.
2. The GUI prototypes used during testing accurately portray their real-world counterparts.
3. The web-enabled mechanism for interacting with the GUI prototypes is not only reliable but also very effective, allowing users to execute tasks in a minimal number of steps.
4. Tab navigation within small-screen digital device GUIs is highly effective and learnable, and is most preferred by users as compared to sliding-icon and button-menu schemes.
5. Bottom-aligned navigation within small-screen digital devices GUIs, though not the most effective or learnable arrangement, is preferred highest as compared to all other edge alignment layouts.

Chapter 2

Design of Experiment Prototype GUIs

2.1 Introduction

An array of design decisions are involved in the creation of a graphical user interface. The use of color, style of icons, and depth of navigation are just a few of the many principles a designer must consider when developing an initial GUI prototype. While some of these design decisions may be relatively easy to make, others involve careful consideration and research. Before proceeding to discuss Web-enabled methods for gauging usability, we must first narrow down the GUI elements that require focus during testing. In this chapter, we describe the process involved in deciding which variable graphical elements to modify during testing, and how these variables were incorporated into the design of the initial SSDD prototype GUIs.

2.2 Variable GUI Elements of Small-Screen Digital Displays

Naturally, the first step in the design of any system is investigating the design of related systems that already exist. The specific small-screen system chosen to analyze for this thesis was the digital camera. Therefore, to gain a sense of existing small-screen GUI designs in this area, five popular digital cameras were placed under inspection: the Sony Cybershot 2.1, HP PhotoSmart C30, Kodak DC240, Polaroid PDC1000, and Fuji MX-2700.

Exploration and comparison of the GUIs for these cameras revealed seven major variable elements, which are summarized as a test matrix in Table A.1-1 of Appendix A and listed in detail below:

- **Widget Types**: A widget describes any screen element that is used to direct a user's navigation. A wide variety of widgets were used among the five cameras surveyed, many of which had shared use (e.g. tabs, icons, and highlighted text were common among the devices). The check mark widget was the only exception to this, being used only by the Polaroid camera.
- **Number of Widget Types on Screen at Once**: At any given time, the number of widget types shown on screen ranged from 1 to 4. Most commonly, two or three types of widgets were used simultaneously.
- **Navigation Alignment**: Due to the small-screen nature of these devices, the tendency of each camera was to display menus and widgets along the edges of the screen, to minimize occlusion of the background image. The cameras used alignment along each of the four edges of the screen.

- **Widget Size**: The sizes of widgets varied from relatively small ($1/30^{th}$ x $1/20^{th}$ of the display size) to relatively large ($1/5^{th}$ x $1/7^{th}$ of the display), depending inversely on amount of functionality presented on the screen at once.
- **Functionality**: Scenario #1 for the functionality variable in Table A.1-1 describes the actions common among all cameras. Some cameras, though, were more sophisticated than the others, allowing the user to control the device's sound, video output, and other bonus features shown in the higher scenarios.
- **Navigation Schemes**: A variety of navigation schemes were employed among the five cameras, ranging from two variations of a tab metaphor, to two variations of a sliding icon metaphor, to a cascading button menu style.
- **Depth of Navigation**: Most commonly, a user could traverse between at most two to three levels of hierarchy during navigation. However, some cameras, such as the Sony Cybershot 2.1, allowed four-level deep navigation.

2.3 Process for Selecting GUI Variable Elements

For an in-depth study into small-screen GUI design, every permutation of all traits within the test matrix could be user-tested to uncover their most usable combination. However, the focus of this thesis is toward a Web-based *tool* for usability evaluation. Therefore, to simplify experimentation, we decided to vary only two of the seven possible graphical elements.

One of the main goals of usability testing is to help give preference to one of two or more difficult, contrasting design choices. Accordingly, when selecting which GUI variables to modify during usability testing, higher consideration should be devoted toward elements that vary the greatest. Among the seven variable elements in the digital camera test matrix, the navigation schemes and navigation alignments were the most unique. A different navigation scheme was used for each camera to guide the user to the device's functionality, and navigation alignments were present on all edges on the screen. As a result, these two elements were the ones chosen as the GUI variables within the experiment.

2.4 Process for Selecting Virtual Device's Mechanical Frame

Although the focus of usability design and testing for this thesis is toward the *graphical* user interface, considerations must also be made toward the layout of the *mechanical* user interface. The number of buttons used, the role of these buttons, and their layout on the device in relation to the screen are all important factors that can affect user evaluation of a GUI's usability. The goal, therefore, in choosing a mechanical design for a GUI-oriented experiment is to select one that's as unobtrusive and simple as possible.

With this in mind, the mechanical layout used during testing was modeled after the Kodak DC240 and Polaroid PDC1000, both highly successful digital cameras for beginners. The final mechanical design chosen is shown in Figure 2-1. To avoid diverting attention from the device's screen, the design is rather simple. As indicated by

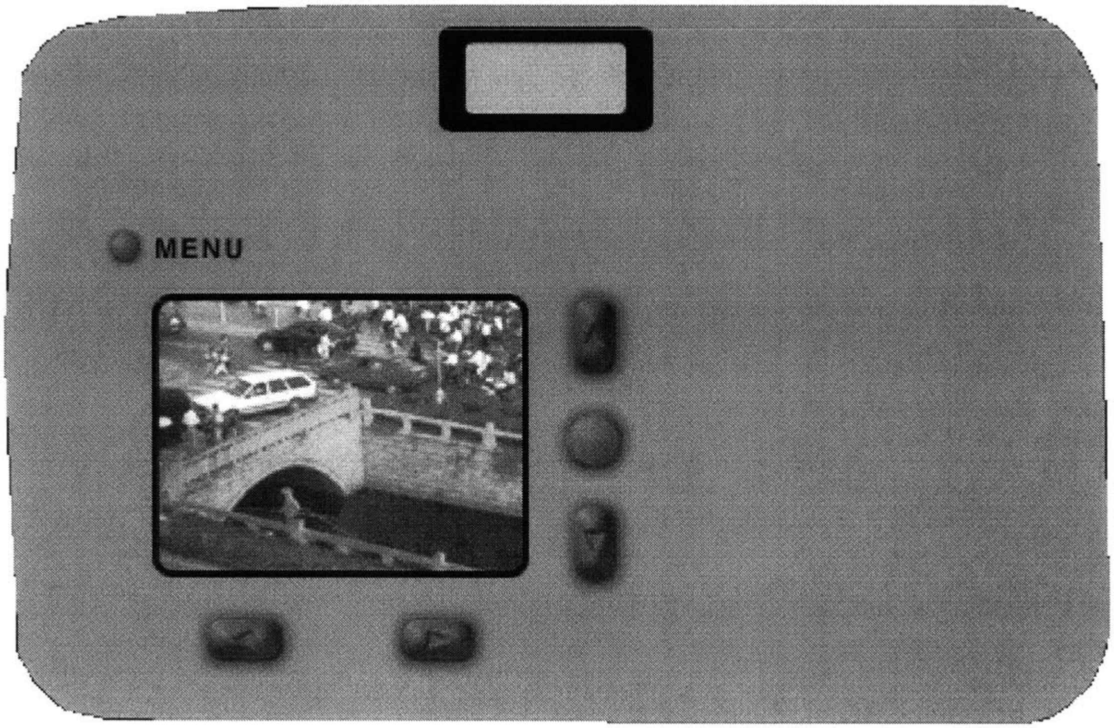


Figure 2-1 Mechanical Design Chosen for the SSDD GUI Experiment

the viewfinder graphic at the top of the device, the user has a view of the back panel of the camera. The LCD screen for the camera is located in the lower-left with the six device buttons situated around it. The button images for the device were given a three-dimensional look so users would know they could push them to interact with the GUI. The Menu button to the upper-left of the screen toggles the GUI menu on and off. Left and Right buttons are positioned below the screen, while the Up, Enter, and Down buttons are placed to the right of the screen.

2.5 Detailed Description of Navigation Scheme GUIs

When conducting an experiment to measure the behavior of a particular variable, it is crucial for the experimenter to fix all other variables so that observed changes in results can be attributed to the one under investigation. Hence, when testing and comparing navigation schemes available for small-screen digital devices, it is important to fix all other variable GUI elements. This section presents a description of each navigation scheme GUI tested for this thesis. For each GUI, Bottom-Aligned navigation was employed, and all other non-scheme elements remained fixed as much as possible to ensure the validity of the results gathered.

2.5.1 Sliding-Icon Navigation Scheme GUI

Both the Kodak and the HP cameras use a "Sliding-Icon" navigation style, also referred to as 'N1' throughout this thesis. For this scheme, navigation starts in category-selection mode by showing a series of boxed-background, functionality-category icons along an edge of the screen. By default, the center icon is highlighted with a bordering text tool tip description. Only a fixed subset (e.g. three) of the available six or more category icons is shown on the screen at one time. The HP camera, for instance, has eight total category icons, but only three are shown on the screen at once. The mental mechanism underlying this style of navigation resembles that of shifting one's view of a circular scroll jutting out from the screen. Initially, the user's view is centered on the icons that can be seen from the "top" of the scroll. In a Bottom-Aligned implementation, as the user navigates right he is in a sense shifting his view of this scroll to the right, sliding the far left icon out of view and uncovering a new icon on the far right, with selection and center-of-view transferred to the right-adjacent icon. Navigation among these icons is cyclical, such that the user can transition from the "last" to the "first" category, and vice-versa.

Settings for each category can be configuring by selecting its corresponding icon, displaying a category-settings menu in place of the original.

Illustrations of the Sliding-Icon navigation scheme are shown in Table A.2-1 of Appendix A.

2.5.2 Button-Menu Navigation Scheme GUI

The Sony Cybershot 2.1, instead, uses a Microsoft Windows® 95-like Button-Menu navigation layout. With this style, navigation starts by displaying three functionality-category, text-only buttons on screen. Selecting these buttons by pressing enter executes a default action for that navigation category. In a Bottom-Aligned implementation, higher levels of navigation within a particular category can be reached by pressing up, which cascades another button menu above the selected category button, allowing a user to change related settings for that category. Menus can, in turn, be hidden by recursing downward.

Illustrations of the Button-Menu navigation scheme are shown in Table A.2-2 of Appendix A.

2.5.3 Tab Navigation Scheme GUI

The Tab navigation scheme, used on both the Fuji and Polaroid cameras, somewhat resembles the button-menu layout except the menu backgrounds are transparent. The Tab scheme is a metaphor that behaves like folders within a filing cabinet, with a folder belonging to each of the camera's functionality-categories. The tabs to all folders are always visible the screen. Initially, a default folder is shown on top, masking all the other folders below it. On each folder, actions relating to the selected category are shown in text, with one action highlighted by default. With a horizontal tab alignment, adjacent category folders are exposed by pressing left or right, causing the text actions of the new category to replace the old text actions. These text actions, then, can be traversed using the up and down camera buttons.

Illustrations of the Button-Menu navigation scheme are shown Table A.2-3 of Appendix A.

2.6 Detailed Description of Navigation Alignment GUIs

This section describes the navigation alignment GUIs used during experimentation. As with the navigation scheme GUIs, all extraneous elements were kept fixed throughout each design to help generate alignment-significant results. Two cameras exercised the Sliding-Icon scheme, though with contrasting alignments (Left-Alignment was used by the Kodak camera, while Top-Alignment was used by the HP camera). As a result, this was the navigation scheme chosen for the alignment GUIs.

2.6.1 Bottom-Aligned Navigation Alignment GUI

With the Bottom-Aligned arrangement, the sliding icons span the bottom portion of the screen, from left to right. Pressing left and right on the category-selection level navigates among the available category icons as described in Section 2.5.1. Pressing up or enter within this level directs a user to the settings menu for the selected category, while pressing down hides the menu. Throughout the settings navigation level, left and right serve to respectively decrease or increase the value of the setting under focus. Selecting up or enter in this level accepts the value of the given setting or initiates an action, whichever is appropriate based on the category stage and option selection. To escape to the view the categories again, a user can press down.

Illustrations describing the Bottom-Aligned navigation GUI can be found in Table A.3-1 of Appendix A.

2.6.2 Left-Aligned Navigation Alignment GUI

With the Left-Aligned arrangement, the sliding icons instead span the left portion of the screen, from top to bottom. The intuition for using this navigation is similar to that for using the Bottom-Aligned style. Button presses directed perpendicular to the edge of alignment advance to the user to higher levels (right) or lower levels (left) of navigation.

In category-selection mode, button presses along the edge of alignment shift the icons shown on the screen. For Left-Aligned navigation, this means that pushing the up button shifts icons down to select the icon directly above the current one, while the down button shifts icons up, placing the icon directly below the current one in focus. Throughout the settings navigation level, up and down serve to respectively increase or decrease the value of the setting under focus. Selecting right or enter in this level accepts the value of the given setting or initiates an action, whichever is appropriate based on the category stage and option selection.

Illustrations describing the Left-Aligned navigation GUI can be found in Table A.3-2 of Appendix A.

2.6.3 Top-Aligned Navigation Alignment GUI

Top-Alignment and Bottom-Alignment are nearly identical. Navigating left or right with both alignments in the category-selection level surveys the available category icons, with left button presses shifting the users view of the "scroll" to the left, and right likewise shifting view to the right. Also, in the category-settings level, left functions to decrease the current setting's value, while right increases it.

Three differences, however, distinguish the Top-Aligned GUI from the Bottom-Aligned GUI:

1) Icons are arranged horizontally across the *top* of screen for Top-Alignment rather than the bottom,

2) The down button *increases* navigation depth rather than decreasing it, and

3) The up button *decreases* navigation depth rather than increasing it.

Illustrations describing the Top-Aligned navigation GUI can be found in Table A.3-3 of Appendix A.

2.6.4 Right-Aligned Navigation Alignment GUI

Right-Alignment and Left-Alignment are nearly identical. Navigating up or down for both alignments in the category-selection level surveys the available category icons, with up button presses shifting the users view of the "scroll" upwards, and down likewise shifting view downward. Also, in the category-setting levels, down functions to decrease the current setting's value, while up increases it.

Three differences, though, distinguish the Right-Aligned GUI from the Left-Aligned GUI:

1) Icons are arranged vertically across the *right* of screen for Top-Alignment rather than the bottom,

2) The left button *increases* navigation depth rather than decreasing it, and

3) The right button *decreases* navigation depth rather than increasing it.

Illustrations describing the Right-Aligned navigation GUI can be found in Table A.3-4 of Appendix A.

Chapter 3

Design of Web-Enabled Usability Tool

3.1 Introduction

When introduced in 1994-1995, the primary asset of the World-Wide Web for site holders was to publish static documents for visitors to read. Over the past five years, however, technology has advanced so far as to allow a site holder much greater possibilities for interacting with Web users. A user is no longer limited to absorbing static content when visiting a Web page, but he is now often able to interact with and even customize the content shown on his browser, as well as communicate back to the maintainer of the site. With a firm understanding of both the GUI designs to be used and the variable elements to gauge during testing, this chapter describes the web-enabled tool developed for performing usability assessment of the small-screen displays for digital devices.

3.2 Usability Testing – What Are We Measuring?

The concept of usability can often be interpreted with different meanings, even among experts within the field. This section outlines the measurable characteristics for usability, and describes those appropriate for measurement in the context of GUI evaluation for small-screen displays.

3.2.1 Measurable Characteristics that Define of Usability

Shackel, in [6], proposes that a usable computer system be portrayed by 'the capability in human functional terms to be used easily and effectively by the specified range of users, given specified training and user support, to fulfill the specified range of tasks, within the specified range of environmental scenarios' (pg. 24). The term 'usability', however, is best understood by describing its quantifiable constituent ingredients. Recognizing this, Shackel proposes four measurable dimensions that define of usability: effectiveness, learnability, flexibility, and attitude.

3.2.1.1 Effectiveness as a Measurable Usability Characteristic

"Acceptable performance should be achieved by a defined portion of the user population, over a specified range of tasks, and in a specified range of environments." [8, pg. 40]

The effectiveness of a graphical user interface, therefore, is largely measured by **the amount of time it takes users to accomplish tasks**, as well **the number of button presses** needed for doing this.

3.2.1.2 Learnability as a Usability Characteristic

"A system should allow users to reach acceptable performance levels within a specified time." [8, pg. 40] In general, as performance level increases within an application, less errors are made. Learnability, therefore, can be measured by **monitoring the number of errors** made by users. If the rate of navigation errors made by GUI users does not decrease significantly starting from initial use, then one can deduce that the interface suffers in learnability. Likewise, if the number of buttons clicked and amount of time taken by a beginning user to accomplish typical device tasks does not reasonably depreciate towards that of an expert user, then learnability is at fault.

3.2.1.3 Flexibility as a Usability Characteristic

"A product should be able to deal with a range of tasks beyond those first specified." [8, pg. 40] Certainly the best way of gauging this characteristic would be to familiarize a user with the initial GUI, gauge his performance, then **add functionality and monitor the user's response**. If the interface is not flexible, then the user will experience a significant decrease in overall performance over time, and effectiveness and attitude of the GUI will noticeably deteriorate.

3.2.1.4 Attitude as a Usability Characteristic

"Acceptable performance should be achieved within acceptable human costs, in terms of fatigue, stress, frustration, discomfort and satisfaction." [8, pg. 40] Although this characteristic not as readily quantifiable as the other ones, it can still be gauged through the use of user evaluations. Test respondents, of course, are the best authority for assessing the degree of costs in using a particular GUI. So, by **asking users to rank attitude-oriented questions/statements on a five-point Likert-style scale**, usability engineers can extract the degree of this cost and thereby gain a quantifiable measure of user attitude toward a GUI.

3.2.2 Usability Characteristics Appropriate for SSDDs

"One should select the dimensions that are appropriate to the users, the particular system and the environment to be considered in specified usability testing circumstances." [7, pg. 20] In the context of a small-screen display application, effectiveness is undoubtedly an appropriate dimension to measure. Small-screen digital devices are generally used on a time-limited basis for a specialized purpose. To qualify as usable, the GUI for these devices should provide a user quick and effortless navigation to the specialized functionality he desires. Therefore, to gauge effectiveness during usability testing, a user's clickstream should be recorded, along with his performance time(s).

Learnability is also important characteristics for screen-screen GUIs. The function of many small-screen devices is to facilitate a range of repetitive tasks for a user. One of the main roles for a PDA device, for example, is to help a user organize his daily activities. Without a high degree of learnability in the graphical user interface, many users would lose their patience when trying to use the device and revert back to traditional non-digital ways scheduling their time, such as manuscripting appointments in a Day-Timer®. For this reason, a user's error rate and performance over time should be monitored to ensure that the time taken and number of buttons pressed per task show reasonable improvement.

Unlike all other characteristics, flexibility is not currently that great a concern in the usability assessment of present-day small-screen graphical user interfaces since most SSDDs, with some exceptions, are geared toward conducting only a fixed, specialized range of tasks. However, as usage demand of these devices increases, and as available functionality increases, we recognize that flexibility becomes a much more important issue. Investigation into flexibility assessment of small-screen GUIs, therefore, is a topic left to future research.

Attitude, on the other hand, is perhaps the most important characteristic to consider in the assessment of small-screen GUIs. As mentioned before, the function of many SSDDs is to facilitate a range a repetitive tasks for a user. If accomplishing these tasks using a SSDD entails greater costs than when performing them manually, a user will likely abandon the digital device. Attitude assessment evaluations, therefore, are necessary during testing.

3.3 Usability Evaluation Techniques

A variety of evaluation techniques exist for gauging the aforementioned usability characteristics. Table 3-1 below, derived from [5, pg. 118], shows the key differences between these methods.

Method	Data & Info	Advantages	Disadvantages
Analytic	Quantitative	Usable early in design. Few resources required, therefore cheap.	Narrow focus. Lack of diagnostic output for redesign. Broad assumptions of users' cognitive operations. Limited guidance on how to use methods, therefore can be difficult for evaluator.
Expert	Qualitative	Overview of whole interface. Few resources needed, therefore cheap. High potential return (detects significant problems).	Restrictions in role- playing. Subject to bias. Problems locating experts. Cannot capture real user behavior.
Observational	Qualitative / quantitative	Quickly highlights difficulties. Verbal protocols valuable source of information. Can be used for rapid iterative development. Rich qualitative data.	Observation can affect user activity and performance levels. Analysis of data can be time-consuming and resource-consuming.
Survey	Qualitative / quantitative	Addresses users' opinions and understanding of interface. Can be made to be diagnostic. Questions can be tailored to the individual. Rating scales lead to quantitative results. Can be used on a large group of users.	User experience important. Low response rates. Possible interviewer bias. Possible response bias. Analysis can be complicated and lengthy. Interviews very time-consuming.
Experimental	Qualitative / quantitative	Powerful method. Quantitative data for statistical analysis. Can compare different groups of users. Reliability and validity good. Replicable.	Time spent on experiments can mean evaluation is difficult to integrate into design cycle. Tasks can be artificial and restricted. Cannot always generalize to full system in typical working situation.

Table 3-1 Overview of the Five Evaluation Methods for Gauging Usability

The two techniques selected for this thesis were *surveying* and *experimentation*. As seen in the table, these methods are capable of acquiring both qualitative and quantitative measures of usability. Specifically, for the usability tool developed in this thesis, Likert-scale surveying was chosen as the desired method for gauging attitude, while experimentation via task completion analysis was chosen for evaluation of the effectiveness and learnability characteristics.

In Likert-scale surveying, a respondent is asked to indicate his level of agreement with a particular statement or question. The number of choices is up to the researcher, but a five-point scale is commonly used.

Task completion analysis, on the other hand, compares the effect of variable elements within a device GUI by monitoring a user's performance over a range of typical device tasks. After undergoing general training on how to use the device, the user is presented a GUI with a particular series of assignments to the variable elements and asked to complete a series of tasks. As the user interacts with the GUI, his button presses and time taken are recorded. The variable element for that stage of testing is then modified and the user is asked to perform the same tasks with the new GUI, again with his performance monitored. After all the different assignments have been made and tested for this variable, a researcher can analyze the collected data for correlations between the user's performance and the GUI element settings.

3.4 Available Hypermedia Technologies

A detailed description of web technologies available for use in the design of hyperactive environments is provided in Chapter 2 of [4]. The information is summarized in Table 3-2 below.

Name	Description	Advantages	Shortcomings
HTML (HyperText Markup Language)	Language that controls the placement content on a web page, including as text, images, and other supported media.	• Very easy to learn and use	• Not very interactive or powerful
CGI (Common Gateway Interface)	The layer that gives the web the ability to transfer information from web clients to programs on the server and then relay the results back to the users.	• Supplies a good amount of interaction between the user and the web page • Can be programmed in nearly any programming language	• Network delays can lead to relatively slow interaction
JavaScript	A lightweight interpreted programming language with rudimentary object-oriented capabilities.	• Enables programmatic control over the appearance of a document's content • Provides control over browser behavior • Allows interaction between the content in a document • Allows the browser to keep track of user actions through the use of cookies	• Not suitable for sole use in complex applications with a large need for interactivity
Java	A language for creating safe, portable, robust, object-oriented, multithreaded, interactive programs.	• Powerful and architeture neutral • Suitable for complex interactive applications	• Use requires knowledge of object-oriented programming

		• Applets can interact with with a browser's JavaScript	
Macromedia Shockwave	Browser plug-in that enables users to play Macromedia Director movies in Web browsers.	• Parent movies can be saved as Java to increase browser compatibility • Able to interact with JavaScript using Lingo, the Director scripting language • An excellent environment for creating complex applications quickly	• Plug-in software available only for Windows® and Macintosh®

Table 3-2 Summary of Available Hypermedia Technologies

3.5 Technologies Appropriate For Survey and Experimental-Based Usability Testing

In deciding which of these technologies to employ toward the creation of a Web-enabled usability tool, it was necessary to consider the parameters that required measurement. To allow Likert-scale surveying, we realized HTML forms and CGI would be needed. In order to ensure that user responses to these surveys were valid, form verification using JavaScript was also essential.

Determining how to implement task completion tests, however, was a much more challenging design decision. Certainly, we realized that we needed to create a series of interactive SSDD prototypes for the user to communicate with, each with varying navigation schemes or navigation alignments. In addition, we realized that each of these prototypes would need to be capable of both recording a user's button presses and monitoring his task-completion time. Though these prototypes could have been implemented using JavaScript alone, we realized a more powerful interaction technology would be more appropriate. The decision, therefore, came down to using either Java or Shockwave. The advantage of implementing the prototypes as Java applets was that the usability test would be available on web browsers for virtually any operating system. We realized, though, that employing Shockwave would greatly simplify prototyping. Using Director, a designer can quickly create an application by copying and pasting screenshots from Adobe Photoshop, and add transitions between these screens with simple Lingo scripts. Using JavaScript-to-Shockwave communication, the designer could record button presses and completion times. Recognizing that many GUI designers would likely prefer using a WYSIWYG application for prototype design over a programming language interface, we decided to use Shockwave applets to display the interactive SSDD prototypes during task-completion testing.

3.6 Use of Technologies in Tool Creation

Screenshots of the resulting SSDD usability tool are shown in Appendix B. The first screen shown is an introduction page, where users are given a brief description of the goal of the experiment, notified what buttons to use on the shockwave-simulated camera, and warned of preparations they may need to make to ensure their results are reliably submitted to our database. When ready to proceed, the user is then led to a User Profile Survey page.

3.6.1 User Profile Survey Page

On the User Profile Survey page, the user is instructed to provide information about himself (name, gender, etc), his educational knowledge, his experience with SSDDs, and his physical disabilities, if any. This information is needed to help analyze trends in user populations from the gathered results. Once all user profile fields have been answered and the user clicks the "Start Experiments!" button, his profile information is saved as a new cookie on his web browser, and he transitions to the first task-testing page.

3.6.2 Prototype Task-Testing Page

The first time the task-testing page loads, the user is shown an alert box that his performance will be monitored throughout the entire experiment. Upon agreeing to this, he then reads the test instructions and begins the first task by clicking the "Start the first task" button. A JavaScript function executes, displaying the first task in the task text box ("Toggle image protection for ONLY the current image"), starting the task-completion timer, and sending a message to the Shockwave plug-in embedded on the page to show the camera's buttons. The user, reading the requested task within the text box, uses the camera buttons to try to accomplish it. While interacting with the GUI, his button presses are recorded within the plug-in. Once the user successfully completes the task for the tested navigation scheme or alignment, his button clickstream is sent to a JavaScript function, which stops the task timer and stores both this value and the clickstream. A series of task-assessment questions may then follow. When done answering these questions, testing of a different task begins. This process continues until the last task is completed and it's assessment questions are answered.

3.6.3 Prototype Evaluation Page

From this point, the task test data from the previous stage of the experiment is saved as a unique cookie on the user's Web browser, and he moves on to a survey page for evaluating the entire navigation scheme or navigation alignment previously encountered. On this page, the user must answer attitude evaluation questions on a four-point scale, from Yes to No. In addition, for each question, a comments box is provided to allow him to explain his rank selection. When completed, this survey data is saved as a unique cookie. Following this survey is a page testing the next navigation scheme or navigation alignment, where the procedure for the first task-testing page repeats.

3.6.4 Overall Impressions Page

After succeeding through the seven task-testing pages and seven evaluation pages, the user is prompted for his overall impressions of the experiment and GUIs encountered. Here he ranks the navigation schemes and navigation alignments from best to worst, indicates to the degree to which extraneous factors, if any, influenced his results, and adds any miscellaneous comments. Again, when complete, this overall impressions information is saved as a new cookie on the user's web browser.

3.6.5 Results Summary and Submission Page

Upon submitting his overall impressions, the user reaches the final page of the experiment, which displays a summary of his experiment results extracted from the saved cookies on his browser. The user clicks the "Submit data (via email)" button, and a CGI program collects all his experiment information sends it as an email to the researcher. The researcher, using a Perl script, can then transfer the data from this and any other experiment responses to a database for later analysis.

Chapter 4

Application of Statistical Analyses to the Hyperactive Usability Environment

4.1 Introduction

In this chapter, we apply a series of statistical tests and analyses to the data variables discussed in Chapter 3. Based on the information resulting from these analyses, we summarize user performance and attitude responses to the test GUIs employed in the hyperactive SSDD Experiment.

4.2 Data Cleaning

A crucial precursor stage to analysis of hypermedia experiment results is data cleaning. Data cleaning is the process of removing polluted information from a database to ensure the reliability of analyses performed on the data therein.

Data pollution can arise in several forms, as indicated in [10]. A common source of pollution is duplication of records. The presence of multiple entries in a database by a particular user, resulting from outright duplication, negligence, or deliberate misrepresentation of information, can corrupt the validity of the ensuing statistical tests. To avoid duplication, each user is required to submit his email address with his experiment results. Before inserting a user's information into the experiment database, we first query for existing records with the same email address. If no records are returned, we assume the user is unique and add his data to the database. Otherwise, upon discovering a matching database entry, the data pending insertion is discarded.

Data pollution also exists in the form of domain inconsistency, which represents information that is not plausible given the domain of valid responses. When prompted to enter his email address, for example, a user may mistakenly neglect his address' domain name (e.g. "tweather" without the "@mit.edu"). Data pollution of this form is primarily prevented by structuring questions in a manner that provides end-users with all possible valid responses. In the SSDD GUI Experiment, for example, most questions require responses through the use of radio buttons and pull-down selection lists. For free-form questions, such as the email address example above, JavaScript verification on the user's Web browser is employed to ensure that responses match the domain of valid answers.

To address all other possible forms of data pollution, data cleaning via database data inspection is performed, once all user responses have been collected. For the SSDD GUI Experiment, a fair amount of extreme outliers were initially visible within the database. These values greatly skewed the original statistical analyses, prompting us to remove many of them from the data set.

One user, for instance, amassed a total number of 100 button presses in attempting to perform a task that required only 4 button presses, drastically increasing the mean number of button presses for this task. Noticing this extreme outlier in the database, we read through the user's comments and analyzed his clickstream, quickly realizing that he had confused the requested task with another one similar in nature. He proceeded to continually execute the wrong task until he finally figured out the true task that needed to be completed. Needless to say, his clickstream and completion time results for this task were discarded, since they obviously did not reflect the effectiveness or learnability of the particular GUI being tested, but rather represented the user's misunderstanding of the directions.

Other users did use a reasonable number of button presses (e.g. 5) during experimentation to accomplish a task, but took excessively long in doing so (e.g. over 3 minutes). In such cases we determined the user was overly idle, perhaps due to an interruption to the experiment. As a result, such outlier time values were discarded, since they did not represent a user's performance working at full capacity with undivided attention.

In the future, real-time outlier deduction during testing should be incorporated into the experiment to avoid the situations described above. For example, after an extended amount of idle time, the experiment could inform the user with the following alert box message and reinitiate task testing: "It appears you have taken a break from testing. Please click OK to restart testing for this GUI." Also, after an exceedingly large number of extended button presses (e.g. 10 times the minimum number), the experiment could inform the user that he appears to confused as to how to accomplish to desired task, and allow him to abandon that particular task test.

4.3 Characterization of the End User

As mentioned in Section 3.6.1, knowledge of certain characteristics as regards the test subjects is essential in analyzing and evaluating their collected results. By segmenting the end users into population groups, we can better assess the strengths and weaknesses of both the hyperactive usability environment and the GUIs themselves.

26 total users were tested with the SSDD Experiment for this thesis. All test subjects had an English reading level above Grade 12, and none were impaired by visual or physical disabilities. The majority of respondents (81%) were aged between 20-39 years of age. A small percentage, however, were younger (7%) or older (11%) than this range. The highest educational level among the participants was split more equally: 1 user had not completed high school, 6 users completed high school but not college, 11 users achieved a bachelor's degree, and the remaining 8 users had attained an advanced college degree. Although everyone tested had some degree of experience with film camera, a considerable fraction of the end user population (27%) had never used a digital camera. From the remaining share of skilled digital camera users, 84% had only novice experience with such devices, suggesting little, if any, prior exposure to the specific

digital camera GUIs used in this SSDD Experiment. A vast majority of users (85%), however, had used a cell phone, PalmPilot, or other type of small-screen digital device.

4.4 Performance Measures of the Experiment GUIs

4.4.1 Grade-Based Overall Performance Evaluation

When analyzing the data gathered using the hypermedia usability tool, a usability engineer will often find it useful to first interpret the combined performance results of the comparative GUI experiments across all users. Statistical data regarding the number of button presses, amount of time taken, and number of errors made by all the users can provide insight as to the overall effectiveness of all the GUIs designed for the experiment, as well as the effectiveness of the mechanism for gathering experiment data. The values for this data are often difficult to interpret when presented in their natural form. To gain perspective on the significance of the gathered information, one would, in addition, like to compare user performance to that of an expert user of the system. By presenting user performance as a 0-100 grade scale as follows,

$$\text{Grade} = \frac{\text{Test User Performance}}{\text{Expert User Performance}} \times 100$$

test user performance can be judged with respect to the minimum/expert values.

A distribution of grades for both task completion times and number of button presses is shown below in Figure 4-1.

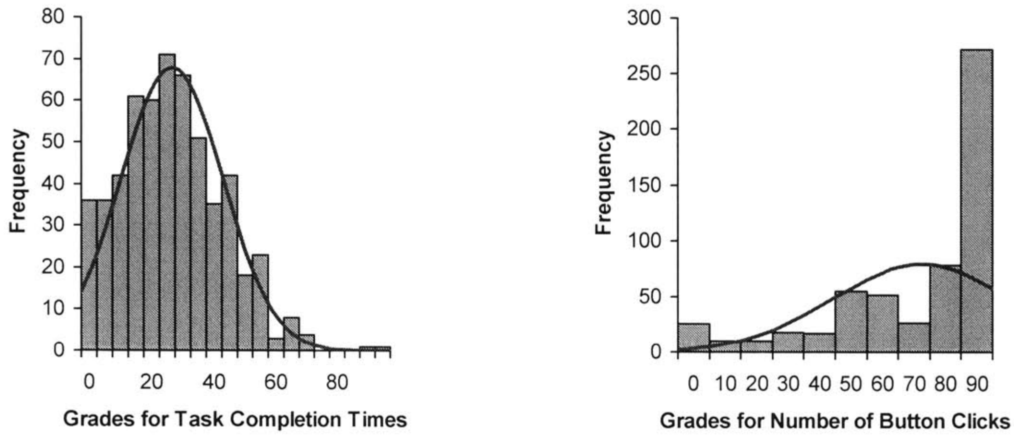


Figure 4-1 Histograms of User Grades with Gaussian Distribution

Completion time performance for the tasks given in this experiment averaged to 29.7% of the expert values, while the number of button presses taken per task by users averaged to 77.5% of the optimal values.

Although 29.7% may seem like an exceedingly low score for task completion times, we should realize that much of the time taken by users is in reading and understanding the task to be accomplished. Upon clicking the Start Task button, an expert user could quickly read each task request and instantly execute it in the quickest manner possible, using the minimum number of button presses. Taking this into consideration, we realize the distribution of task completion times follows what would be expected. In rare instances, users could very quickly read a task requested of them and could execute it in time comparable to that of an expert. This describes the grades shown above 80% in the distribution. More commonly, though, some degree of time would be associated with a user both understanding the task to accomplish and figuring out how this task is executed with the new GUI presented to him. The performance values for the task completion time, therefore, have a normal distribution about the average score.

77.5% is a very reasonable average score for button click performance. As shown by the scores between 90% and 100% in Figure 4-1, a large proportion tasks were accomplished using the minimum or very-close-to-minimum number button presses. Occasionally, however, users were either confused by the task description or got lost within the GUI, leading to extraneous button presses during task navigation and, in turn, poorer performance.

4.4.2 Comparative Performance Evaluation

While an overall evaluation of all performance data may be helpful during analysis, the main focus of the experiment is comparative evaluation of the GUIs tested. Described in this section are comparative performance analyses of the navigation schemes and alignments.

While interpreting the time-completion and button-press performance discoveries presented here, it is important to recognize the possible existence of transfer effects across the GUI tests in this experiment. As Bailey describes in [11], "People change in the course of a study. Experience gained while performing changes them, even if only slightly." That is, practice with a particular task trial could naturally benefit or detriment user performance on a subsequent trial. The GUIs and tasks tested throughout the SSDD Experiment were presented in the same order to all users. Naturally, practice with the earlier GUI tests would serve to affect performance on later trials. In future deployment of hyperactive usability experiments, transfer effects should be controlled by randomizing the presentation of GUI tests to the experiment participants.

4.4.2.1 Completion-Time Performance Evaluation for the Navigation Schemes

Shown below in Figure 4-2 and Table 4-1 are comparative performance measurements for the time taken to accomplish the various navigation scheme task tests used throughout the experiment.

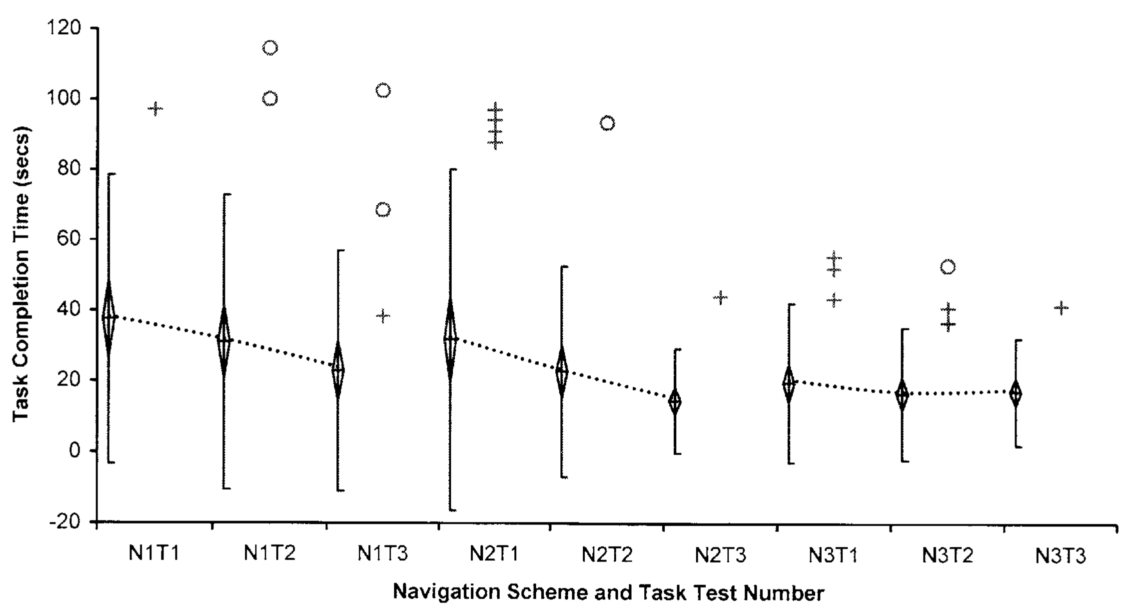


Figure 4-2 Graph of Comparative Task-Completion Times for the Navigation Scheme Tasks

Task ID	Mean Task Time	StDev
N1T1	37.6415	24.88463
N1T2	31.1541	25.36059
N1T3	23.0333	20.68852
N2T1	31.8276	29.34159
N2T2	22.7938	18.16293
N2T3	14.4957	8.96862
N3T1	19.5421	13.63593
N3T2	16.4912	11.33138
N3T3	17.0614	9.17611

Table 4-1 Statistical Data of Task-Completion Times for the Navigation Scheme Tasks

As shown by the dotted lines in Figure 4-2, there is a learning curve associated with using each of the three navigation schemes; As task experience with each navigation scheme grows, the user's completion times decrease. The learning curve for the Tab navigation scheme (N3T1-N3T3) is noticeably less steep than that of the other two schemes; Performance for the Tab navigation scheme decreases by only about 2.5 seconds on average from the first to third task, as compared to 14.5 and 17 second differentials for the Sliding-Icon and Button Menu schemes. This observation from the task-completion times suggests the Tab scheme as most learnable one of the three schemes tested.

Browsing through Table 4-1, we also notice that the Tab scheme has the lowest task-completion mean times. This suggests the Tab scheme as the most effective one of the three schemes tested.

4.4.2.2 Button-Press Performance Evaluation for the Navigation Schemes

Shown below in Figure 4-3 and Table 4-2 are comparative performance measurements for the number of button presses taken to accomplish the various navigation scheme task tests used throughout the experiment, presented in grade-based format.

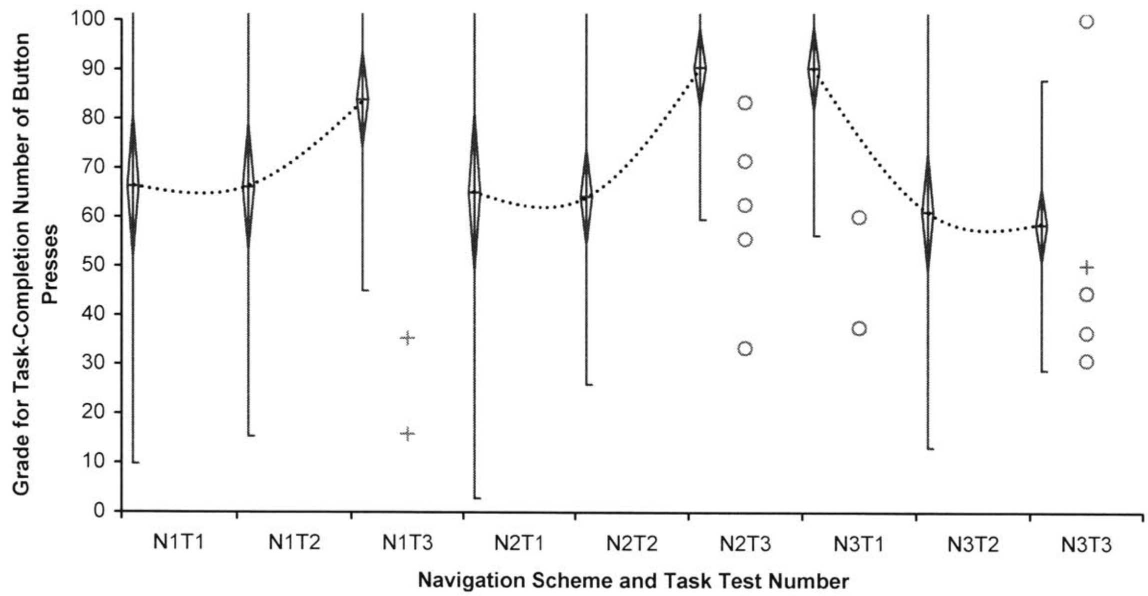


Figure 4-3 Graph of Comparative Grades for the Number of Task-Completion Buttons Pressed During Navigation Scheme Task Testing

Task ID	Mean Grade for Number of Button Presses	StDev
N1T1	66.36	34.400
N1T2	66.17	30.933
N1T3	83.90	23.681
N2T1	65.05	37.849
N2T2	64.20	23.283
N2T3	90.36	18.762
N3T1	90.20	20.627
N3T2	60.98	29.106
N3T3	58.31	17.949

Table 4-2 Statistical Data of Grades for the Number of Task-Completion Buttons Pressed During Navigation Scheme Task Testing

Again, the dotted lines illustrate a learning curve associated with each navigation schemes. Unlike the previous completion-time graph, Figure 4-3 shows grade-based performance, which explains why the curves point tend to arc upwards. The learning curves for the Sliding-Icon and Button-Menu schemes are almost identical to one another: each curve first experiences little growth between the first and second tasks, then each experiences rapid growth between the second and third tasks. The curve for Tab menu, however, actually experiences significant *negative* growth. That is, button-press performance *decreases* with increasing practice. Upon analyzing the gathered clickstream data for the Tab scheme tests, we notice that many users neglected to recognize the two navigation shortcuts available for accomplishing the last two requested tasks with the minimum number of button presses, leading to the decreasing values in performance. As experience with the camera grows, we realize that these shortcuts are likely to be found and utilized by many more users, providing improved effectiveness.

Contrary to the task-completion time findings, the graphs and statistical data for grade-based button-press evaluation of performance suggest that the Sliding-Icon and Button-Menu schemes are essentially equal in learnability and effectiveness, and that both surpass the Tab scheme in these areas of usability.

4.4.2.3 Completion-Time Performance Evaluation for the Navigation Alignments

Shown below in Figure 4-4 and Table 4-3 are comparative performance measurements for the navigation alignment task tests used throughout the experiment.

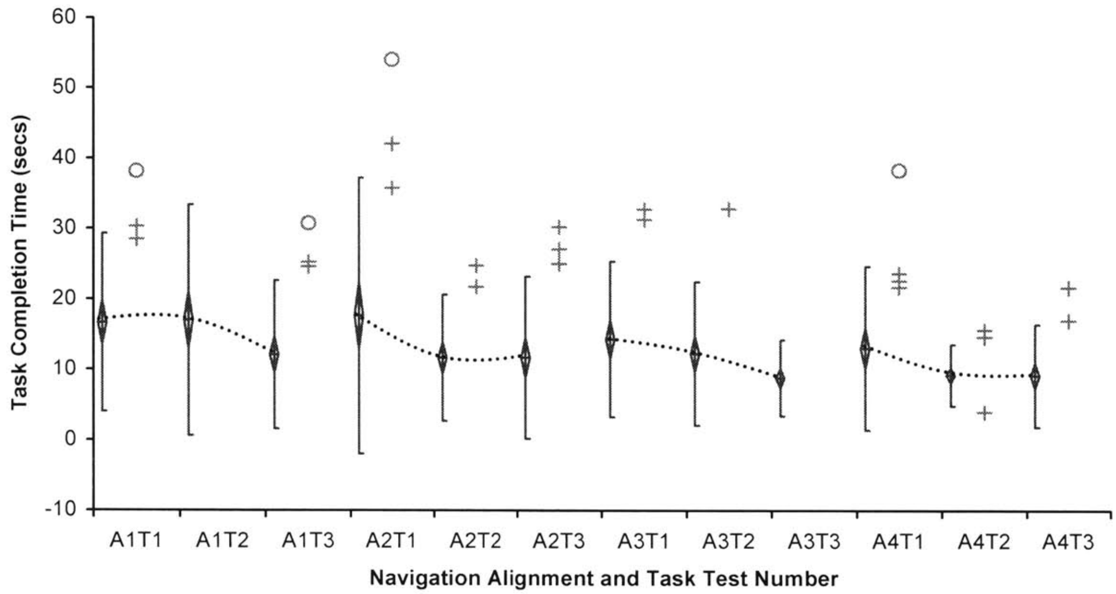


Figure 4-4 Graph of Comparative Task-Completion Times for the Navigation Alignment Tasks

Task ID	Mean Task Time (secs)	StDev
A1T1	16.6762	7.69102
A1T2	17.0386	9.97537
A1T3	12.1117	6.39579
A2T1	17.6496	11.93405
A2T2	11.6307	5.45905
A2T3	11.6985	7.01160
A3T1	14.2992	6.72195
A3T2	12.2500	6.19100
A3T3	8.8060	3.28227
A4T1	12.9861	7.05423
A4T2	9.1645	2.63538
A4T3	9.0817	4.41994

Table 4-3 Statistical Data of Task-Completion Times for the Navigation Alignment Tasks

The dotted lines in Figure 4-4 show the learning curves for the four navigation alignments. The curve for Right-Alignment levels off the quickest for the four layouts.

The differences in steepness among the curves is so slight, however, that each should truly be considered equally learnable.

The statistical data shown in Table 4-3 suggest Right-Alignment as the most effective layout, with completion time values all below 13 seconds. Top-Alignment falls into second place, Left-Alignment places third, and Bottom-Alignment takes last place. The chronological descension of these alignment rankings, however, strongly hints the interference of transfer effects.

4.4.2.4 Button-Press Performance Evaluation for the Navigation Alignments

Shown below in Figure 4-5 and Table 4-4 are comparative performance measurements for the navigation alignment task tests used throughout the experiment.

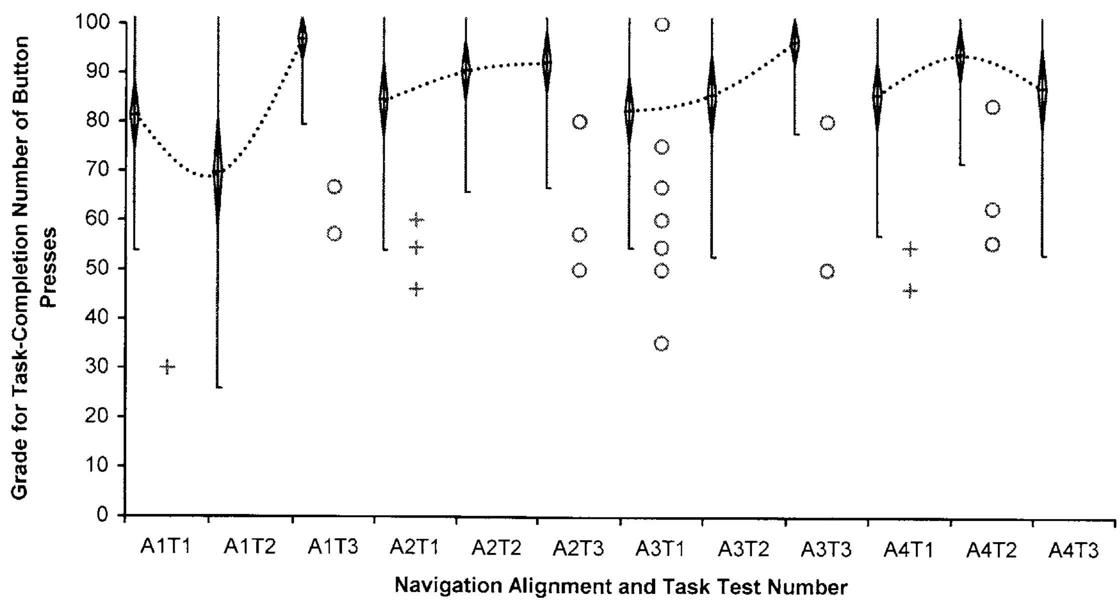


Figure 4-5 Graph of Comparative Grades for the Number of Task-Completion Buttons Pressed During Navigation Alignment Task Testing

Task ID	Mean Number of Button Presses	StDev
A1T1	81.31	16.687
A1T2	69.70	26.631
A1T3	96.95	10.637
A2T1	84.55	18.606
A2T2	90.48	15.011
A2T3	92.17	15.601
A3T1	82.23	16.893
A3T2	85.43	19.912
A3T3	96.25	11.349
A4T1	85.49	17.321
A4T2	93.89	13.555
A4T3	86.88	20.494

Table 4-4 Statistical Data of Grades for the Number of Task-Completion Buttons Pressed During Navigation Alignment Task Testing

The dotted lines in Figure 4-5 illustrate the variety of learning curves for the four navigation alignments. Observation of these button-press paths reveals abnormal curvature for the Bottom-Alignment (A1T1-A1T3), with performance first decreasing, then increasing drastically. The most reasonable explanation for this occurrence is user difficulty in understanding or executing Task #2 ("Start the slideshow with a 7 second interval") in a Bottom-Aligned GUI implementation. With a 10-25% improvement in mean button-press performance for A1T2, the learning curve for Bottom-Alignment would show normal form.

The curve for Right-Alignment (A4T1-A4T3) also shows abnormal form, though of an opposite nature: performance first increases, then decreases. The source of this irregularity is probably user difficulty in understanding or executing Task #3 ("Navigate into the Brightness menu and accept the current brightness level") in a Right-Aligned GUI implementation. With a 7-13% improvement in mean button-press performance for A4T3, the learning curve for Right-Alignment would show normal form.

With the current navigation alignment data set and button-press performance measures, Left-Alignment boasts the least steep, normal form learning curve, suggesting this menu layout as the one with the highest learnability. Right-Alignment follows closely in second place in learnability.

Multiple 90%+ performance values plus a mid-80% performance value for Left-Alignment mark it as likely the best menu layout in terms of effectiveness. Top-Alignment and Right-Alignment tie for second place in effectiveness with very similar button-press grade performances. Bottom-Alignment performance on task #2 is relatively poor compared to the other alignments, deeming it the last place finisher in terms of effectiveness.

4.5 Attitude Measures of the Experiment GUIs

As mentioned in Section 3.2.2, attitude is likely the most crucial usability characteristic to gauge in the assessment of SSDDs. With this in mind, a number of navigation scheme and navigation alignment-specific questions were gathered from Evaluating Usability of Human-Computer Interfaces [9] and added to the prototype evaluation pages as described in Section 3.6.2 and 3.6.3. In addition, overall user impressions of the different navigation scheme and alignment GUIs were collected on the final evaluation page of the experiment, providing rank information among each of these GUI styles. In this section, we present the results from both the question-based and rank-based evaluations, and incorporate relevant comments from users to draw conclusions on user attitudes.

4.5.1 Question-Based Attitude Assessment for Navigation Schemes

4.5.1.1 Evaluation of Task-Specific Questioning

After completing each navigation scheme task, users were asked two task-specific attitude-gauging questions:

1. Does the sequence of activities required to complete this past task follow what you would expect?
2. Is it clear what stage the camera has reached during your navigation for this past task?

The numerical results from these evaluation questions are illustrated in Figure C.1-1 and Figure C.1-2, and the statistical data is enumerated in Table C.1-1 and Table C.1-2 of Appendix C. From these results, we can reason that a little more than "Most of the Time" (2.00), the sequence of activities for completing each of the tested tasks for each of navigation schemes follows what would be expected, and that for each it is reasonably clear what stage the camera has reached during navigation. We notice that for Task 1 ("Toggle image protection for ONLY the current image"), the Sliding-Icon (N1) navigation scheme seems to be relatively unpredictable (2.19) and confusing (2.15) when compared to the Tab (N3) navigation scheme (2.69 and 2.65). With regards to the predictability of the Sliding-Icon scheme, in fact, one user remarked, "Sometimes it's difficult to anticipate what screen will appear next." We also observe that the sequence of activities for executing Task 2 ("Start the slideshow with a 3 second interval") is relatively unpredictable for the Button Menu (N2) scheme when compared to the sequence for executing the other tasks for this GUI style (2.12 vs. 2.31 and 2.68). The confidence backing these conclusions, however, is not particularly strong given the current set of user data. Sampling a larger number of people in the experiment would likely provide additional confidence to support these interferences, as well as expose other noteworthy conclusions regarding the task-specific strengths and weaknesses of each GUI navigation scheme.

4.5.1.2 Evaluation of Prototype-Specific Questioning

During the prototype evaluation stage of the experiment, the user has finished task testing for the current navigation scheme and is asked the following prototype-specific attitude assessment questions:

1. Does information appear to be organized logically on the screen? (e.g. menus organized by probable sequence of selection, or alphabetically)
2. Are the navigation menus clearly separated from the image background?
3. Is the method of selecting options (e.g. navigating menus and selecting actions) consistent throughout the system?
4. Does navigation using this "Sliding Icon" style feel natural?

The numerical results from these evaluation questions are illustrated in Figures C.1-3 through Figure C.1-6, and the statistical data is enumerated in Table C.1-3 through Table C.1-6 of Appendix C. From these results, we can reason several other user-attitude trends toward the three available navigation schemes. Looking at Figure C.1-4 and Table C.1-4, for example, one can see that in the opinion of the users tested, the navigation menus for the Tab scheme are considerably less distinct from the background image (2.08) than for the other schemes (2.58 and 2.65). Also, Figure C.1-5 and Table C.1-5 reveal a possible deficiency in the Button-Menu navigation scheme as it relates to the consistency of selecting options throughout the system (2.23 vs. 2.62 and 2.46), a problem similar to our earlier discovery of this scheme's relative unpredictability. One user commented on way to improve this weakness:

```
[For the Button Menu scheme] the 'file' menu is set up
differently than the 'preferences' & 'actions' menus.
When you click on file; you should see only format.
Selecting format should then bring up the options
menu-'protect this', 'protect all', etc.
```

In terms of naturalness as regards to "look-and-feel," Figure C.1-6 and Table C.1.6 of Appendix C indicate that users seem to prefer the Tab scheme (2.35 vs. 1.96 and 2.04). This tendency is expected, considering how closely the Tab layout resembles it's real-world, file-cabinet-folder counterpart as it relates to categorizing information.

The Sliding-Icon scheme may be considered unnatural by some users since there are, in a sense, two ways of rationalizing navigation for this layout. As described in Section 2.5.1, there are two navigation object layers for the Sliding-Icon metaphor: a square highlighted view area and a scroll of icons. One user might initially imagine that navigation controls the movement of the view area in relation to a fixed-position scroll of icons, as is currently the implementation, whereas another user might imagine the opposite scenario, that of moving the scroll of icons behind a fixed view area. An analogy to this situation would be the use of a mouse for navigation in a flight simulation computer game; some users feel it's more natural to pull the mouse toward them for pulling the plane up, much like how a lever device works, whereas other users feel more natural pushing the mouse away from them to raise the plane's nose.

People tend to associate the Button Menu layout with Microsoft® Windows® 95 and desktop displays, which may explain why this scheme is considered less natural than the Tab scheme. As one user notes, "[The Button Menu scheme] seems weird for a handheld device. Looks a lot like Windows."

4.5.2 Rank-Based Attitude Assessment for Navigation Schemes

In the final stage of evaluation, users were shown screenshots of the navigation schemes used throughout the experiment and asked to provide overall ranks among them. The results are shown as histograms in the below table.

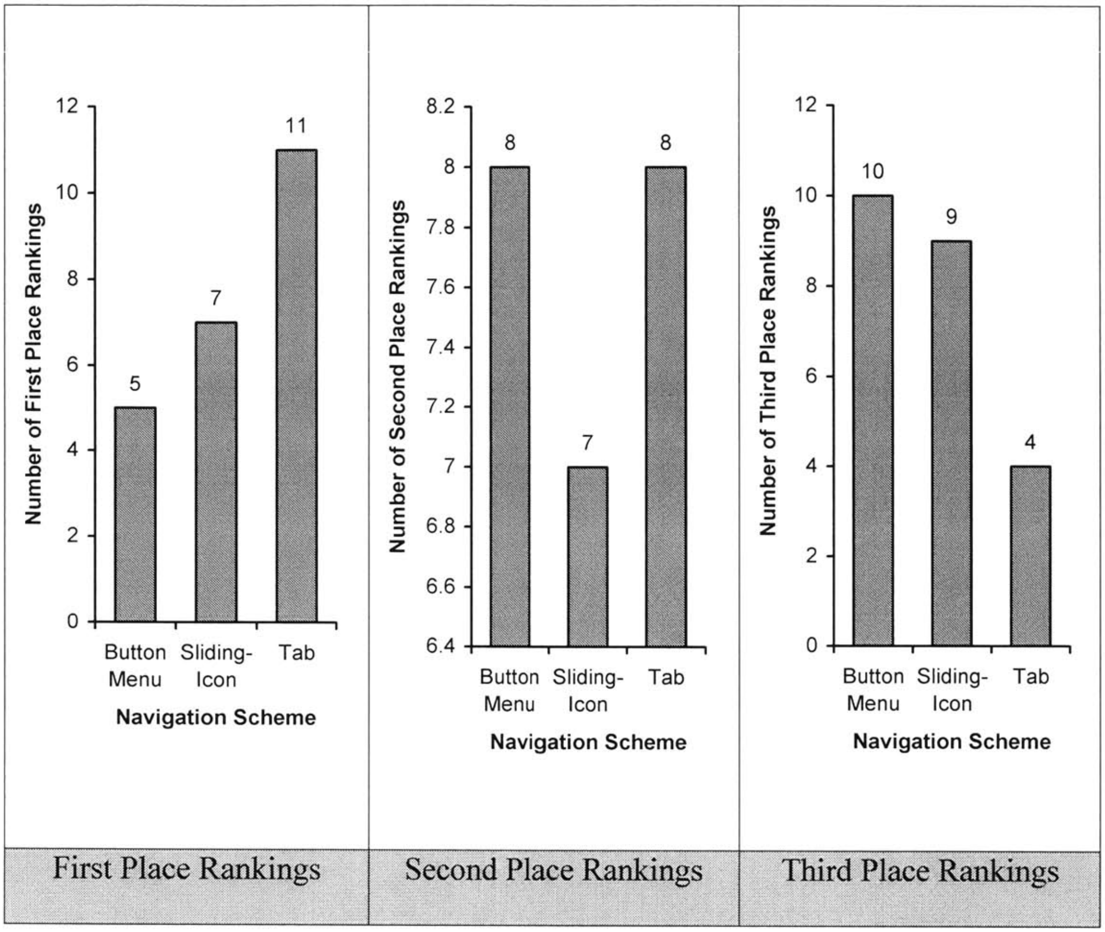


Table 4-5 Navigation Scheme Rankings

From inspection of the first and last rankings alone, we see consistent staircase patterns: Preferences increase for each passing scheme and, likewise, dislikes decrease for each passing scheme.

Assuming that preference among the navigation schemes is linear, the below figure illustrates rankings in a summative manner, with first place ranks assigned 3 points each, second place rankings given 2 points, and third place rankings awarded 1 point.

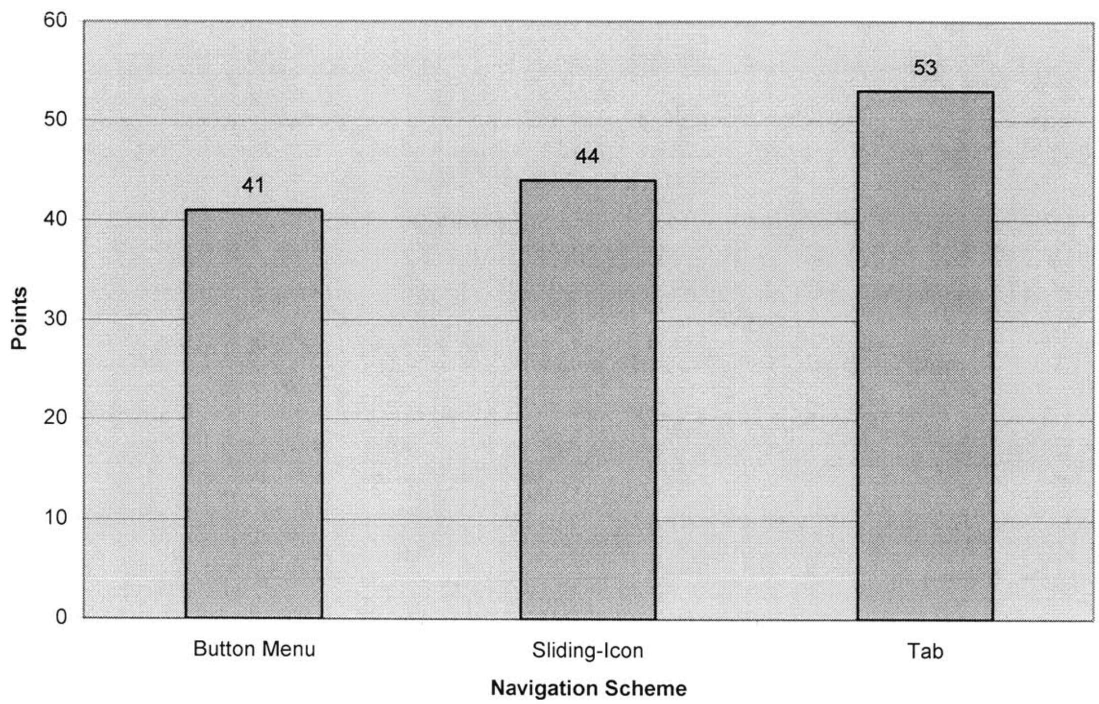


Figure 4-6 Summative Navigation Scheme Rankings

As you can see from both Table 4-5 and Figure 4-6, the Tab scheme is clearly ranked the highest and, therefore, is the most preferred among all users.

4.5.3 Question-Based Attitude Assessment for Navigation Alignments

During the prototype evaluation stage for the navigation alignment portion of the experiment, the user is asked the following two overall attitude assessment questions:

1. Are the menu options aligned properly on the screen?
2. Does the alignment (bottom/top, left/right) of the navigation menus feel natural?

The numerical results from these evaluation questions are illustrated in Figures C.2-1 and Figure C.2-2, and the statistical data is enumerated in Table C.2-2 and Table C.2-2 of Appendix C. Despite the moderately small number of people surveyed for the experiment, we notice definite patterns in users' attitudes toward the four different alignments layouts. Users deem Bottom-Alignment (A1) an exceedingly proper means for the layout of menu options on the screen (2.96), and they feel very natural using it for navigation (2.81). Left-Alignment (A2), on the other hand, is regarded as relatively improper (2.42) and exceptionally unnatural (1.69). Opinions for Top-Alignment (A3) are divided between the two attitude assessment questions; while users are convinced that the menu options are aligned properly on the screen for this layout (2.62), they are generally opposed to the belief that Top-Alignment is natural (1.65). This same division of attitudes is echoed for Right-Alignment (A4), though to a lesser degree (2.62 and 2.12).

When reading over the comments for these alignment-specific questions, we realized that the mechanical framing of the virtual device had a significant effect on user attitude. As shown in Figure 2-1, the buttons for the device simulation were arranged with the left/right buttons below the screen, and the up/enter/down buttons located to the right of the screen. This layout of buttons, in conjunction with the Sliding-Icon scheme, influenced many users to favor the Left and Right Alignments, since these GUIs provided a somewhat direct relation between up/enter/down button presses and navigation/selection on the screen. Comments supporting this conclusion are shown in Table 4-6 below.

Alignment Type	Comments
Left-Alignment	"This is a lot easier to use than the horizontal arrangement. I didn't need the horizontal buttons at all!"
Left-Alignment	"I like this better, no reason to use more buttons if three work fine"
Left-Alignment	"I like this set up better than the bottom. Action prompt opposite the accept button."
Right-Alignment	"This is the easiest to use of all the ones I've seen. I like how the menu is close to the buttons."
Right-Alignment	"This was easier because I only had to use the bottom/top buttons plus the menu was right next to the buttons which made it easy to associate them too."
Top-Alignment	"Dot [enter] button doesn't make as much sense in this one, vertically aligned worked better."

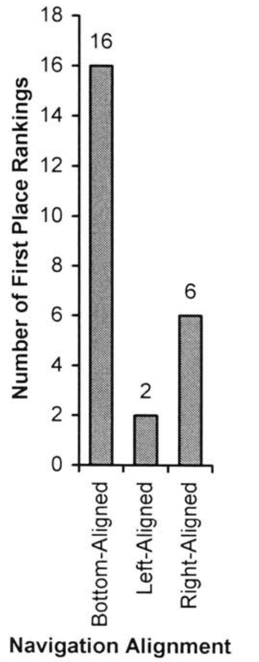
Table 4-6 User Comments Validating the Influence of Mechanical Design on Alignment Attitude

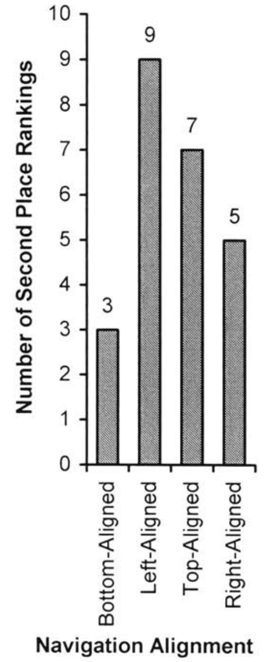
One objective in this thesis was to select a mechanical design for the experiment that was as unobtrusive and simple as possible, as described in Section 2.4. In this manner, information extracted from the experiment could be attributed largely to the graphical user interface rather than the mechanical user interface. However, with most devices, it is often impossible to avoid any sort of ergonomics interplay between these two design facets. The conclusions derived from alignment testing for this experiment prove that the layout of buttons in relation to the display for an SSDD play an important role in a user's assessment of its GUI, and this relation should be kept in mind during graphical design.

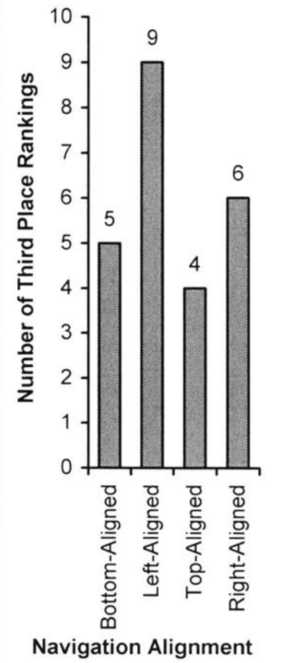
Despite the trend toward positive assessment of the Left and Right-Aligned GUIs, users still had strong preference to the Bottom-Aligned GUI, due largely to their experience using the Start bar for navigation on the Microsoft® Windows® 95 desktop operating system. For the first attitude assessment question (properness), one user points out, "It feels weird for the menu to be at the left of the screen horizontally. I prefer it at the bottom going vertically. People are more used to menus being vertical, I guess from using computers." A similar comment was collected for the second alignment assessment question (naturalness): "I'm used to having menus at the bottom. At the top feels weird. I guess from using Win95 so much."

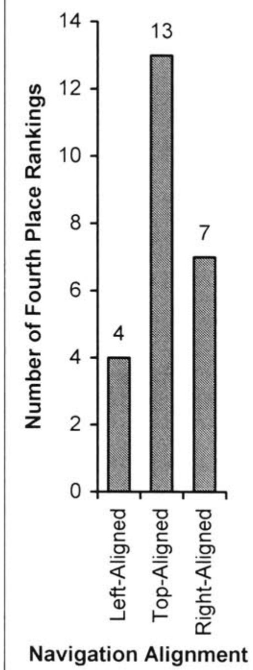
4.5.4 Rank-Based Attitude Assessment for Navigation Alignments

In the final stage of evaluation, users were shown screenshots of the navigation alignments used throughout the experiment and asked to provide overall ranks among them. The results are shown in the below histograms:









First Place	Second Place	Third Place	Fourth Place

Table 4-7 Navigation Alignment Rankings

Several details stand out from the above table. First, we notice that Top-Alignment was never selected as the best alignment, and Bottom-Alignment was never selected as the worst alignment. Also, we observe that more than half the users surveyed chose Bottom-Alignment as the best menu layout, while more than half also chose Top-Alignment as the worst layout. Given that menus on the screen will be arranged horizontally, these results strongly suggest a designer to choose Bottom-Alignment over Top-Alignment.

Assuming that preference among the navigation alignments is linear, Figure 4-7 below illustrates rankings in a summative manner, with first place ranks assigned 4 points each, second place rankings given 3 points, third place rankings awarded 2 points, and fourth place rankings allotted 1 point.

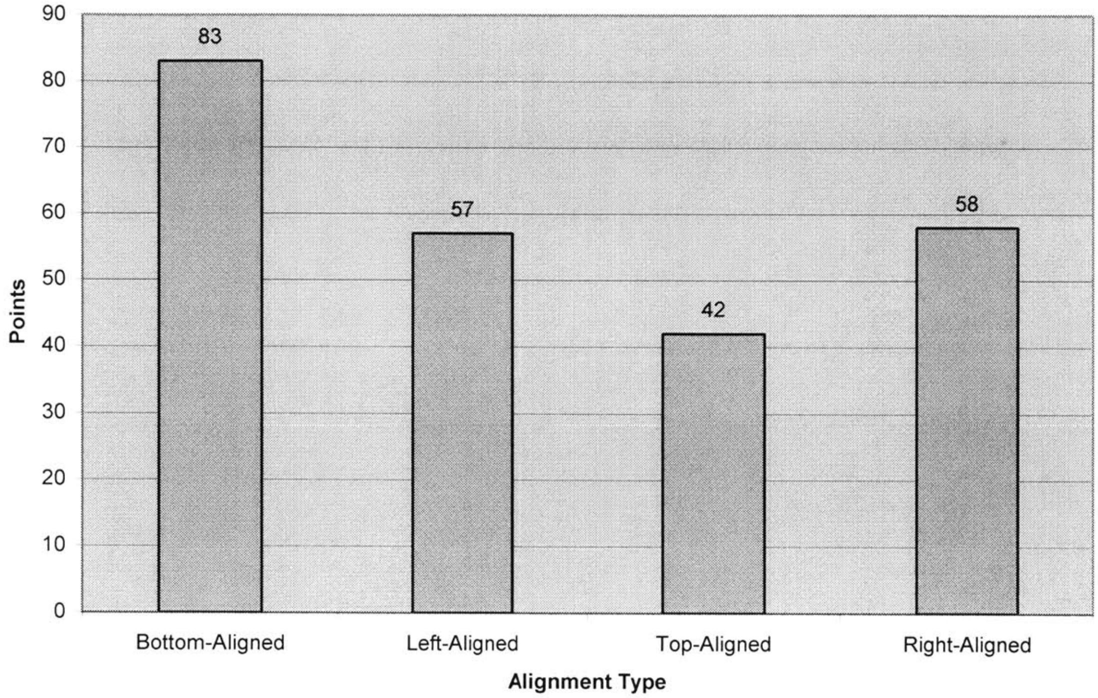


Figure 4-7 Summative Navigation Scheme Rankings

As you can see above, Bottom-Alignment is clearly ranked the highest and, therefore, is the most preferred among all users. Left and Right-Alignment are essentially tied for the second and third place slots, while Top-Alignment falls into last place.

Chapter 5

Conclusions and Further Work

As described in Chapter 1, the primary focus of this thesis was to first present a framework for designing and analyzing web-based, hyperactive usability environments, then introduce and assess a web-enabled usability tool designed according to this framework. The secondary goal of this thesis was to apply this tool toward evaluating the use of sample variable elements within the graphical user interface for a small-screen digital device, so as to help discover guiding design principles within this realm and, in turn, advance the usability these devices.

Based on statistical analysis of data gathered from the Small-Screen Digital Device GUI Experiment, the following conclusions can be made:

1. Although users, on average, executed at less than a third of the expert performance as it relates to their task-completion times, button-press performance during task testing was nearly optimal. This indicates that while some degree of time was needed to read and understand each requested task, the GUIs used during testing and the mechanism for interaction with these prototypes were very effective, allowing users to execute tasks in a minimal number of steps.

2. Comparative evaluation of navigation schemes suggests Tab navigation as the most learnable and effective navigation scheme as it relates to task completion-time performance, while Sliding-Icon and Button-Menu navigation qualify as most learnable and effective as it relates to task button-press performance. With increased exposure to the Tab scheme, navigation shortcuts are likely to become utilized by more users, thereby increasing the effectiveness of this style.
3. The navigation menus for the Tab scheme are considerably less distinct from the background image than as compared to the other schemes. Users, however, feel this style to be the most natural in terms of look-and-feel. Overall, this scheme is preferred highest by users.
4. The Sliding-Icon navigation style is fairly unpredictable and confusing when judged against the other two schemes. Users rank it overall as the second best of the three tested.
5. Despite its resemblance to the Windows® 95 style of navigation, users felt the method for selecting options using this scheme was relatively inconsistent. This scheme was preferred least by users.
6. All navigation alignment layouts are equally learnable with regards to completion-time performance, but Left-Alignment is most learnable as defined by button-press performance. Right-Alignment is the most effective layout as it relates to completion-time assessment, while Left-Alignment, on the other hand, boasts this designation when comparing button-press grades.

7. Users deem Bottom-Alignment an exceedingly proper means for laying out navigation menus, and they feel very natural using it for navigation. This alignment is the most preferred overall among all users.
8. Left-Alignment is regarded as relatively improper and exceptionally unnatural. Users rank it overall second best.
9. Users consider Right-Alignment highly natural and mostly proper. It essentially ties with Left-Alignment for second place in overall ranking.
10. While users are convinced that menu options are aligned properly on the screen for Top-Alignment, they don't think of it as very natural. On the whole, this alignment was deemed the worst of the four.

As described in the Introduction, the utility of the hypermedia usability tool should be measured based on three attributes: reliability, content, and worthiness. In terms of reliability, performance and attitudes for the variable GUI elements tested in the SSDD Experiment closely resemble those gathered from informal user examinations of the parent digital cameras. As demonstrated by the scores of charts and figures presented in Chapter 4, this implementation of the hypermedia usability tool provides a wealth of relevant usability information from which to derive conclusions. Despite how reliable the numerical results may be or how much information can be gathered, the ultimate evaluation of this tool rests on the number of meaningful conclusions that can be extracted from the collected data, and how well these conclusions can be applied toward improving the prototype's GUI design. Based on the useful conclusions derived from the SSDD Experiment data, as outlined in the previous paragraph, we recognize numerous usability strengths and weaknesses among the various navigation schemes and alignments, all of which suggest areas of improvement in the design of real-world small-screen digital device GUIs.

Through the use of the methodology described in Chapter 1 and the techniques applied throughout this thesis, better design and analysis can be employed to improve the utility and effectiveness of web-based usability-assessment environments as shown for the Small-Screen Digital Device GUI Experiment. Data mining analyses using Mann-Whitney and Chi-squared tests, for instance, would likely reveal valuable non-parametric correlations between test user populations and their experimental usability results. In this manner, small-screen graphical user interfaces could then be tailored during design toward the appropriate target audience group.

The hypermedia usability system can be further extended in a number of ways beyond its current implementation. As is, GUI prototype interaction is accomplished using Shockwave applets, restricting the test audience to Windows® and Macintosh® clients. Interaction via Java applet-implemented GUI prototypes is an option that should be explored to provide users architecture-independent access to the hypermedia usability experiment. Additionally, the usability tool can be supplemented with a Web portal to the experiment database to provide administrators universal access to user test records.

The philosophy of the Hypermedia Teaching Facility is that a balance between actual and simulated experimentation maximizes the information provided to GUI designers during usability testing. With the advance of Internet technologies, however, web-based experimentation may very well become the mode of choice in the future.

Appendix A: GUI Prototyping Tables and Figures

This appendix contains graphs and tables relevant to the design of the initial GUI prototypes for the hyperactive usability experiment.

A.1 GUI Test Matrix

Variable GUI Elements	Scenario #1	Scenario #2	Scenario #3	Scenario #4	Scenario #5	Scenario #6	Scenario #7
Widget Types	Tabs (Fuji, Polaroid)	Icons (All)	Buttons / Message Boxes (Sony, Fuji)	Text Highlight (Kodak, Polaroid, Fuji, HP)	Scroll Bar (Kodak, Sony)	Arrows (Kodak, Sony)	Check marks (on/off) (Polaroid)
Number of Widget Types on Screen at Once	1 (Sony, Polaroid)	2 (Sony, Polaroid, HP)	3 (Fuji, Kodak, Sony)	4 (Sony)			
Navigation Alignment	Bottom-Horizontal (Sony, Polaroid)	Left-Vertical (Kodak)	Top-Horizontal (HP)	Right-Vertical (Polaroid)			
Widget Size	1/10, 1/5x1/7	1/24, 1/12, 1/5, 1/10, 1/20. 1/8	1/5x1/20, 3/8x1/8	1/9x1/11, 1/30x1/20	1/10x1/6		
Functionality	Capture, Delete, Timer, Review, Flash	Slideshow, Thumbnail, Zoom	Protect, Quality Select, Format	Quick Review	Language Selection, Date & Time	Video, Exposure, LCD Brightness	Sound
Navigation Schemes	Tabs w/ text selection (Fuji, Polaroid)	Horizontal sliding icons (HP)	Vertical sliding icons w/ text selection (Kodak)	Icon selection, Tab w/ text selection (Polaroid)	Button-Menu (Sony)		
Depth of Navigation	2 (HP, Kodak, Fuji, Polaroid)	3 (Sony)					

Table A.1-1 Test Matrix of Variable GUI Elements

A.2 Navigation Scheme GUI Diagrams

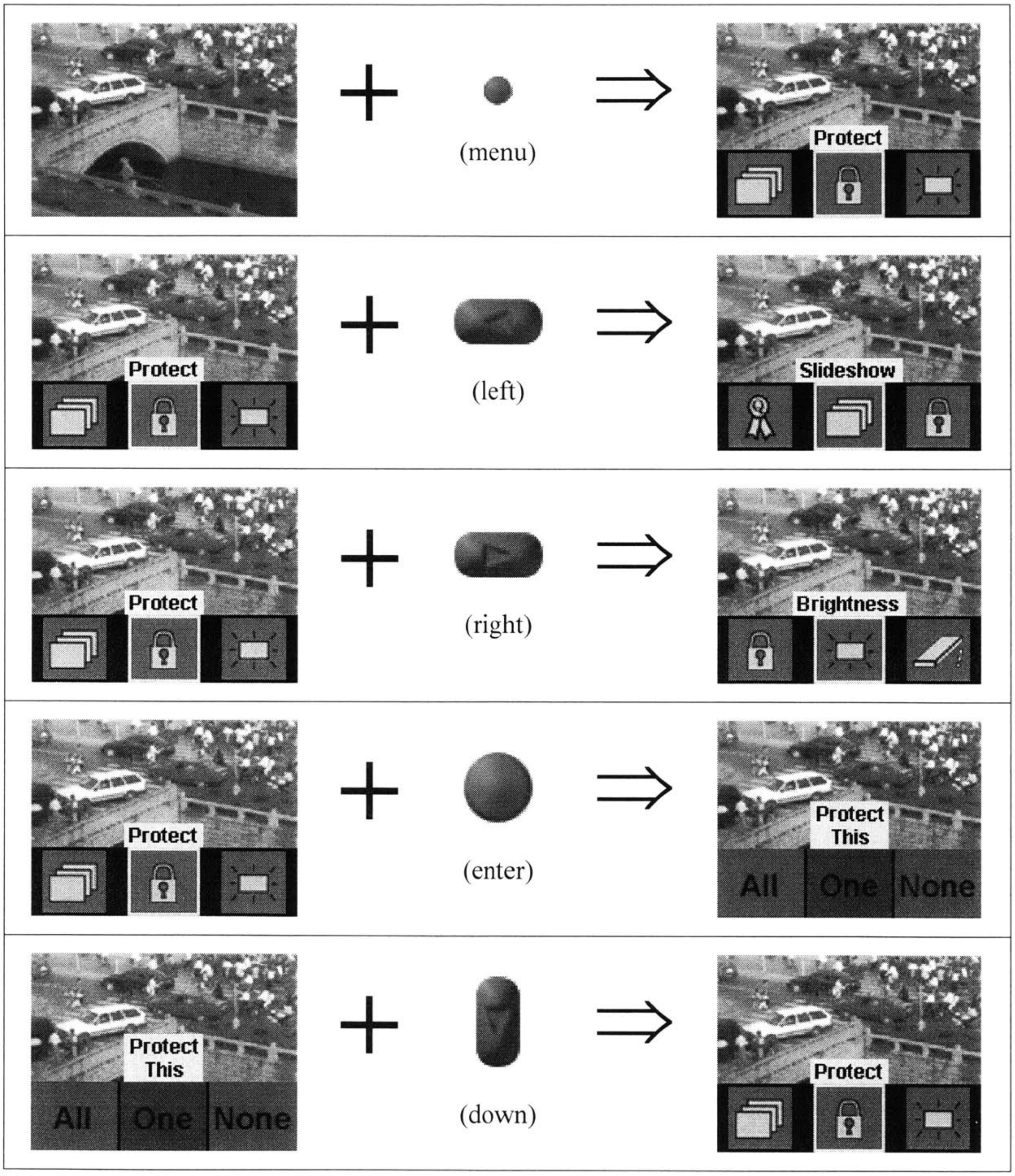


Table A.2-1 Diagrams Illustrating the Sliding-Icon GUI Scheme

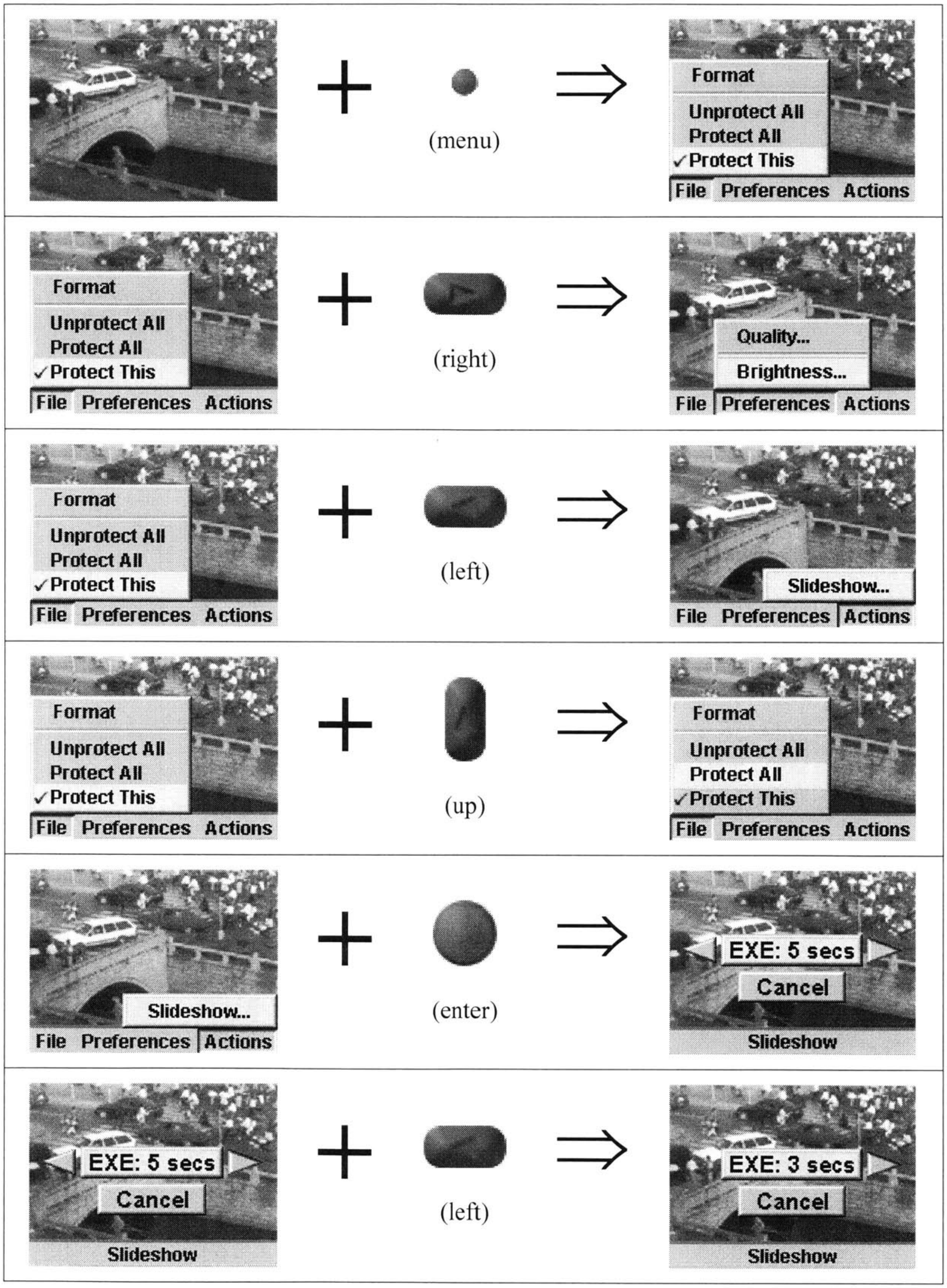


Table A.2-2 Diagrams Illustrating the Button-Menu GUI Scheme

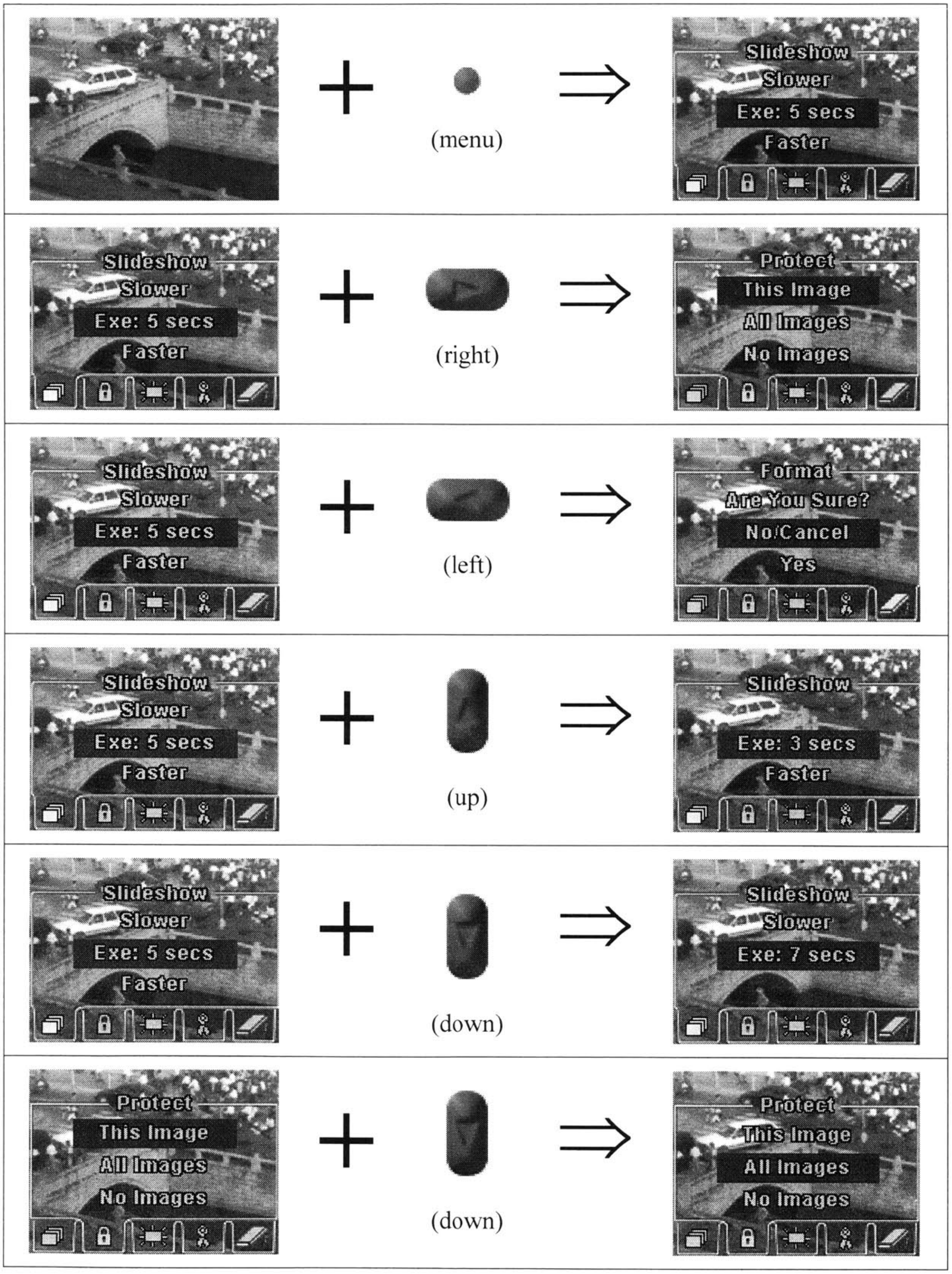


Table A.2-3 Diagrams Illustrating the Tab GUI Scheme

A.3 Navigation Alignment GUI Diagrams

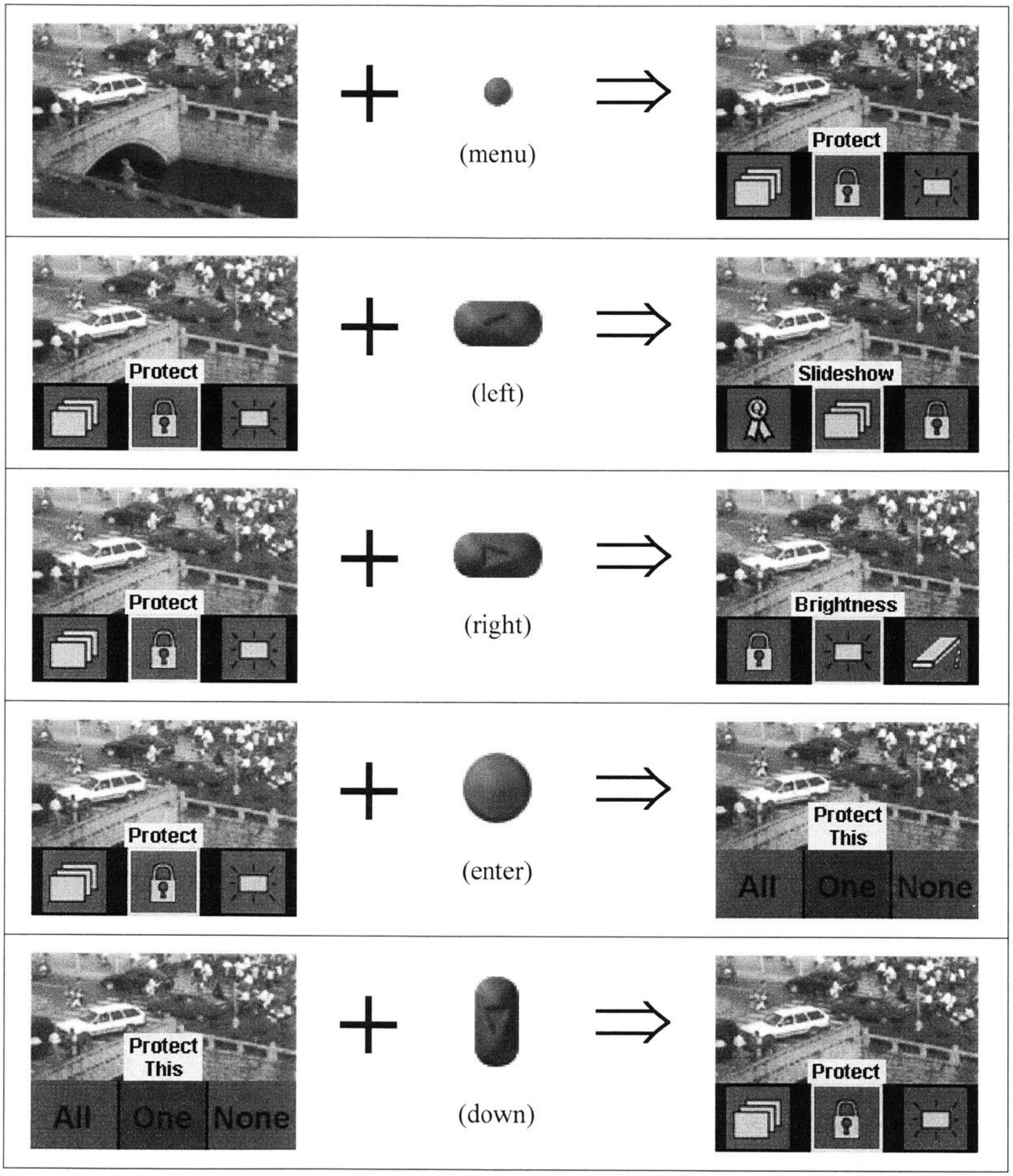


Table A.3-1 Diagrams Illustrating the Bottom-Aligned GUI Scheme

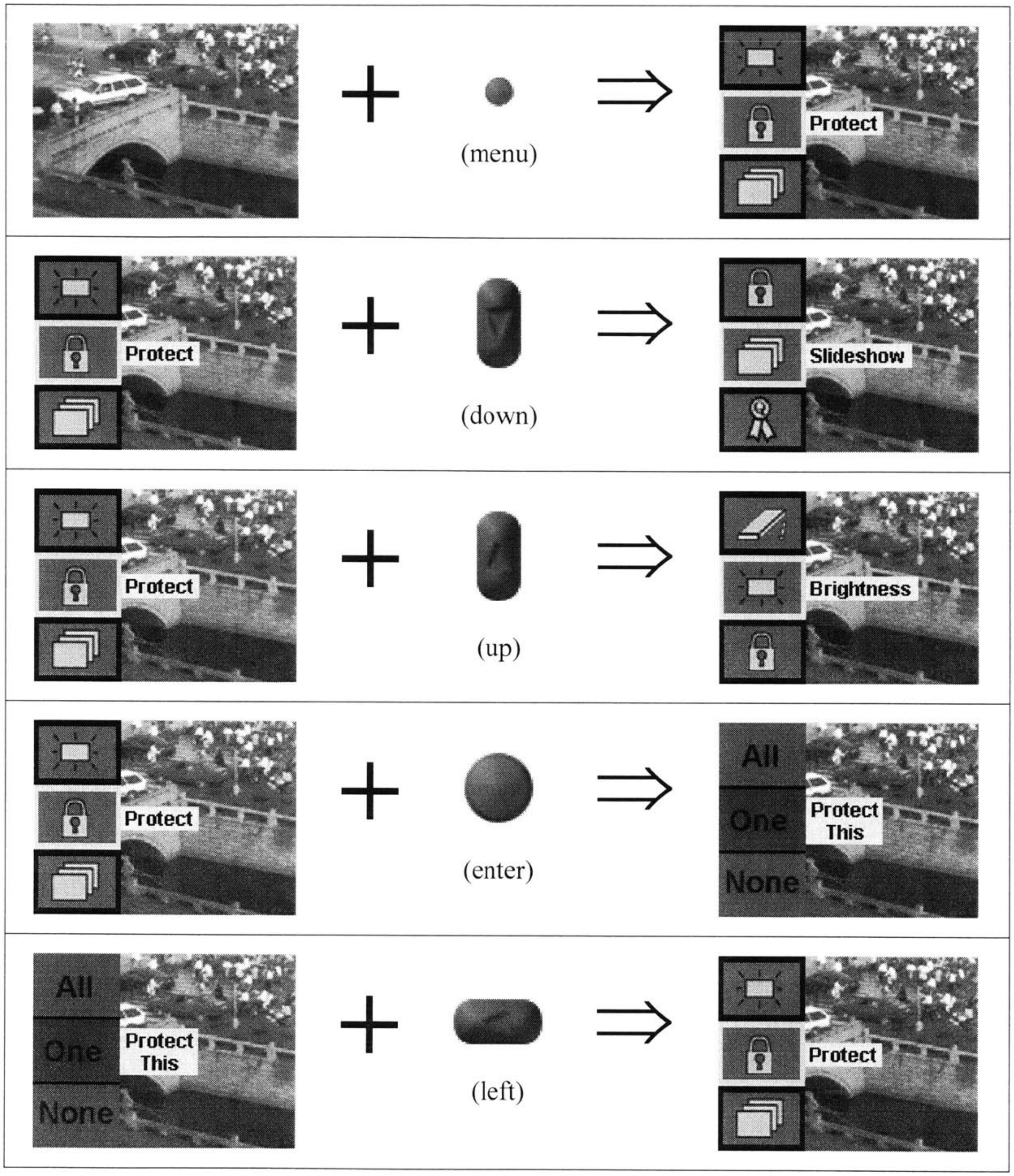


Table A.3-2 Diagrams Illustrating the Left-Aligned GUI Scheme

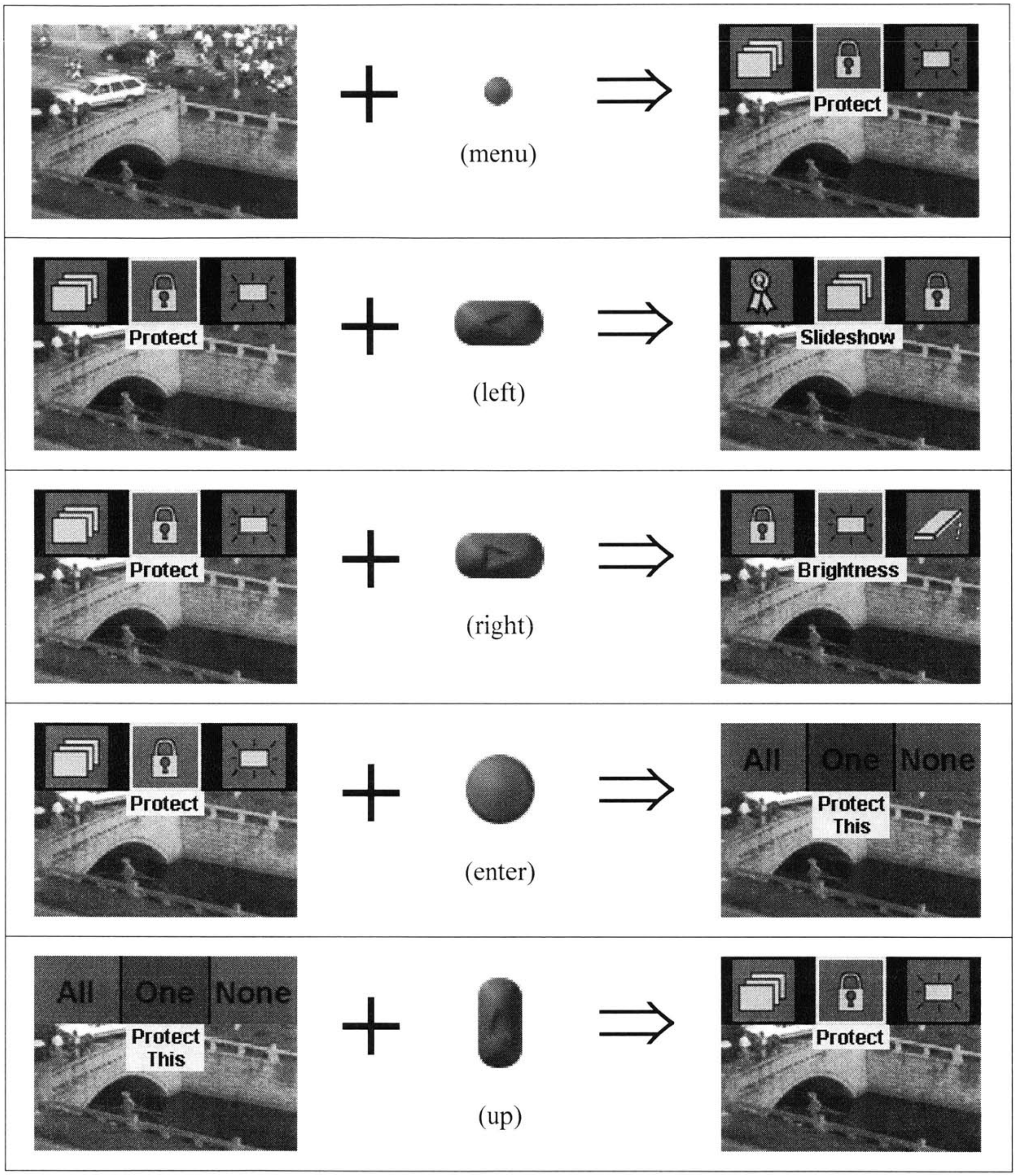


Table A.3-3 Diagrams Illustrating the Bottom-Aligned GUI Scheme

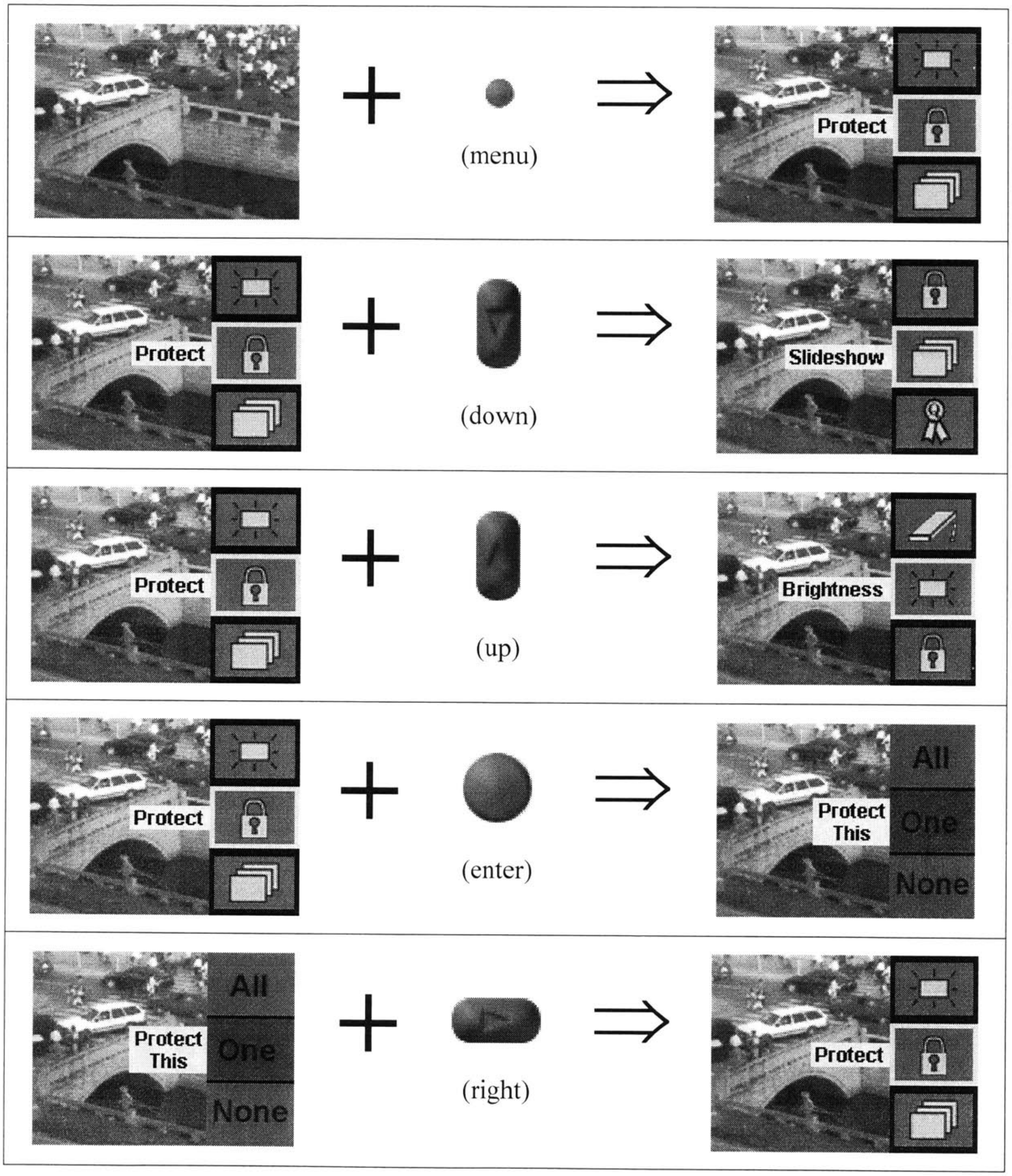


Table A.3-4 Diagrams Illustrating the Left-Aligned GUI Scheme

Appendix B: Screenshots of the SSDD Usability Tool

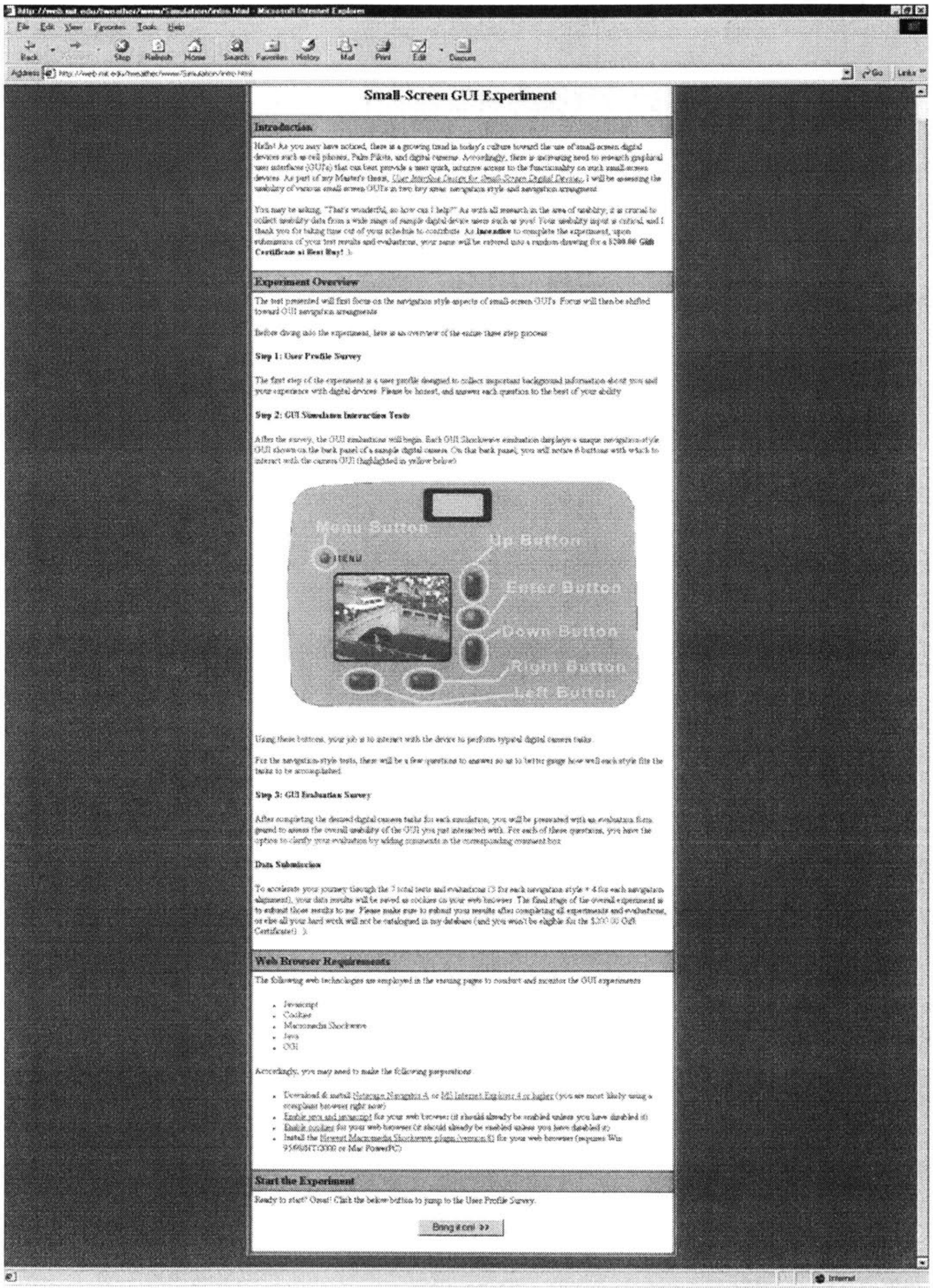


Figure B-1 Experiment Introduction Page

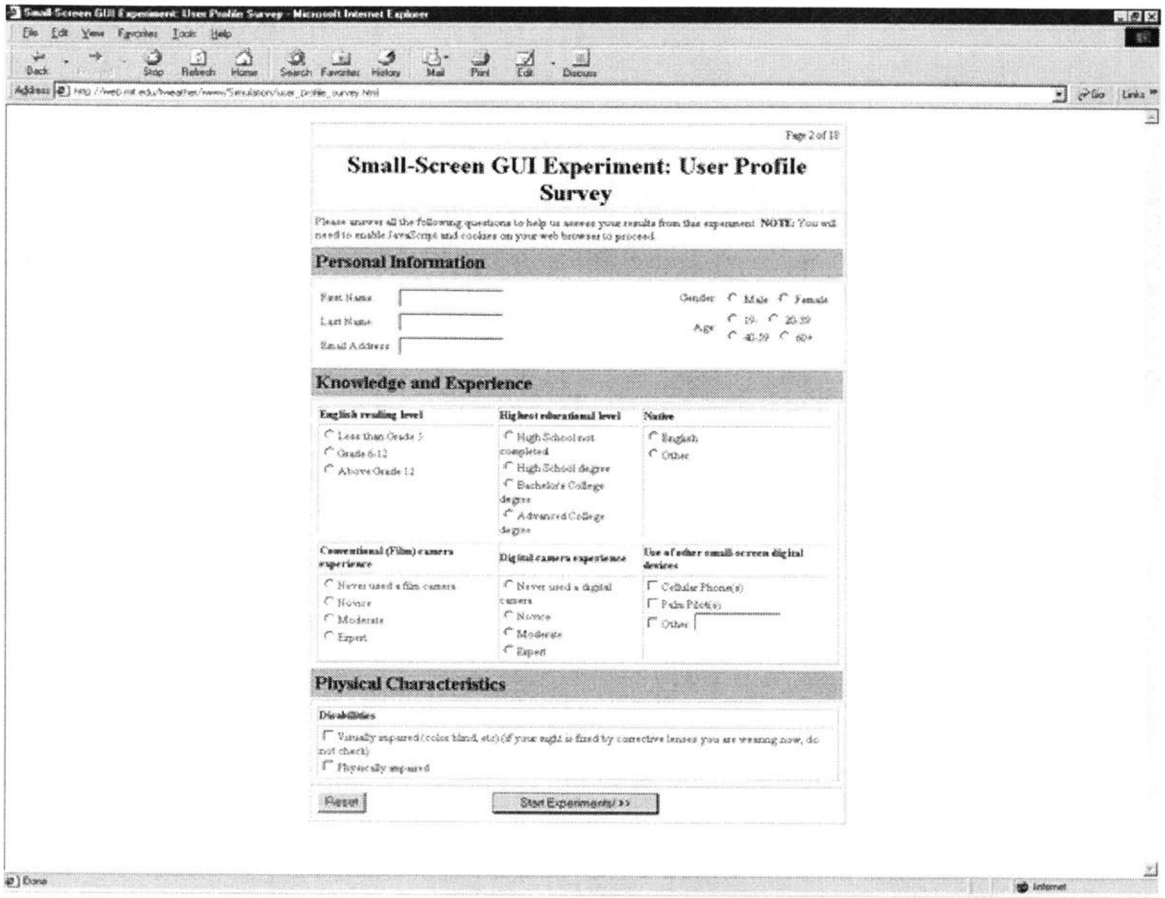


Figure B-2 User Profile Survey Page

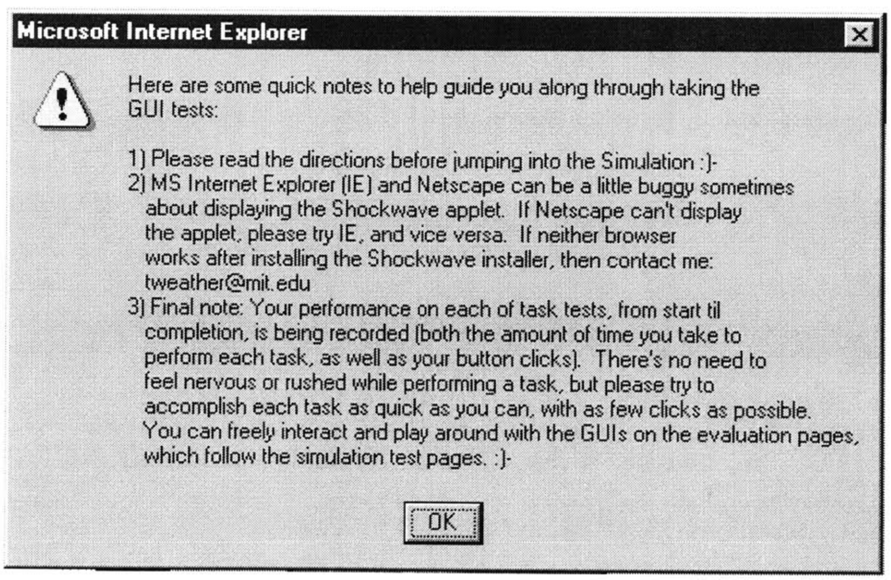


Figure B-3 Alert Box Shown Upon Entrance To the Initial Task-Test Page

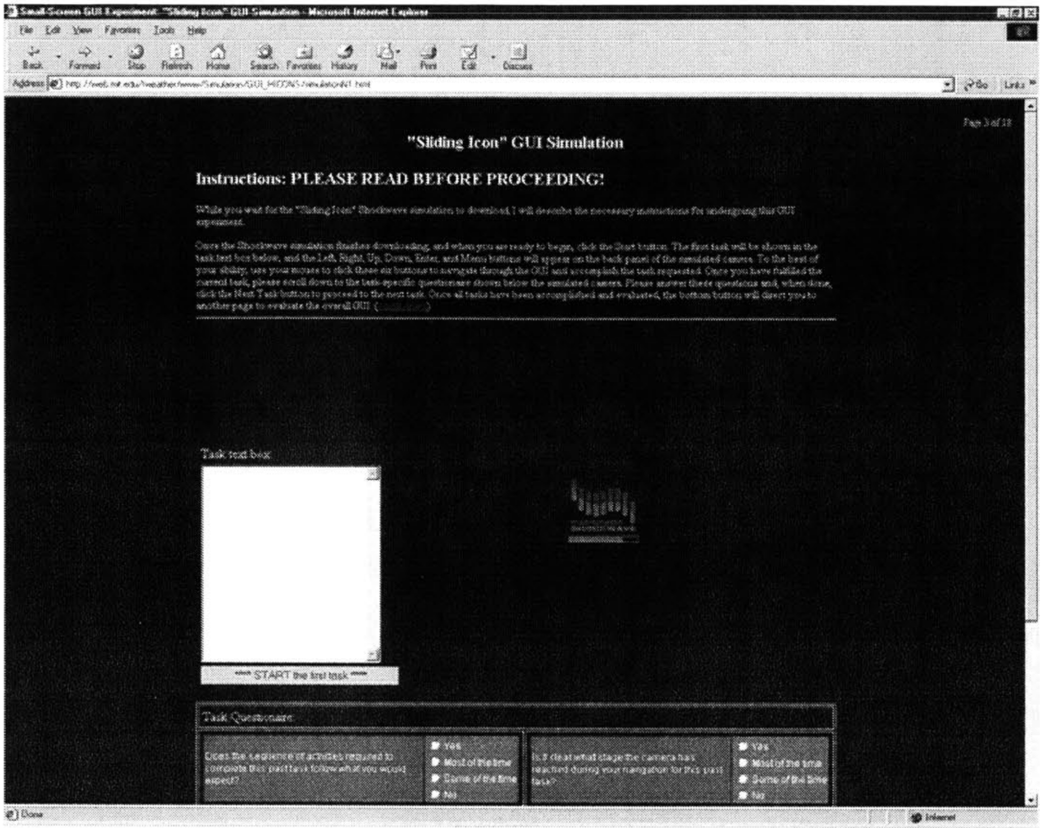


Figure B-4 The Shockwave Prototype Downloading on the Task-Testing Page

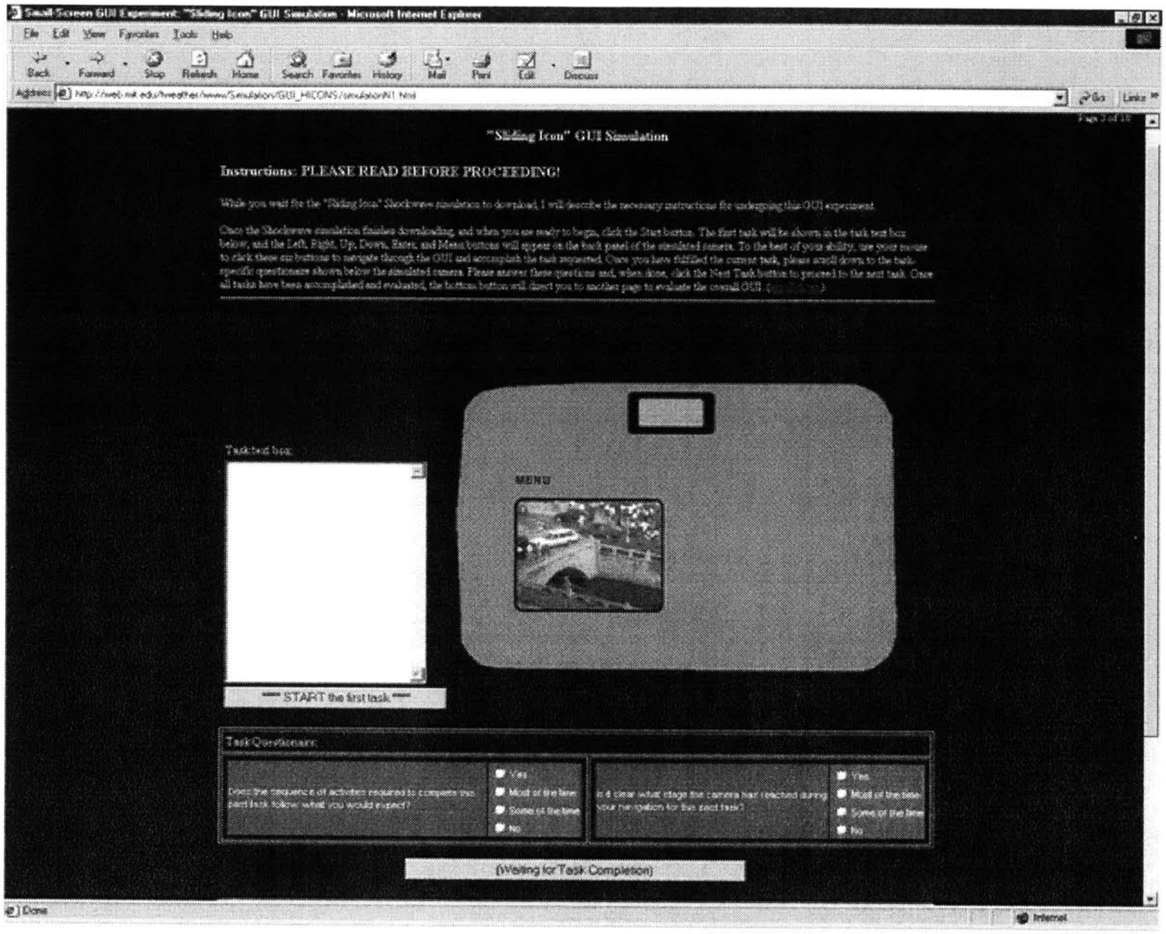


Figure B-5 The Shockwave Prototype Shown After Being Downloaded

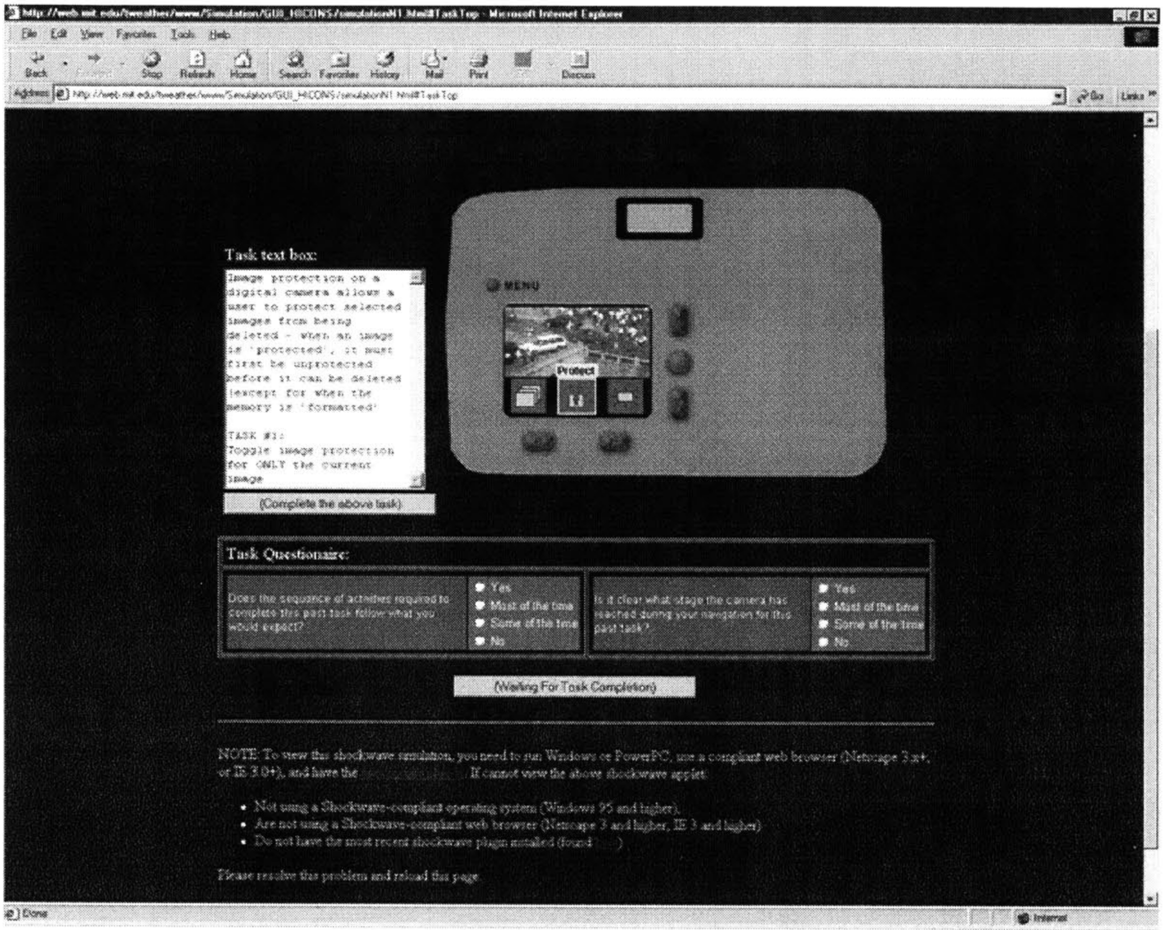


Figure B-6 Shot of Screen During Testing of the First Task for the Sliding-Icon Scheme

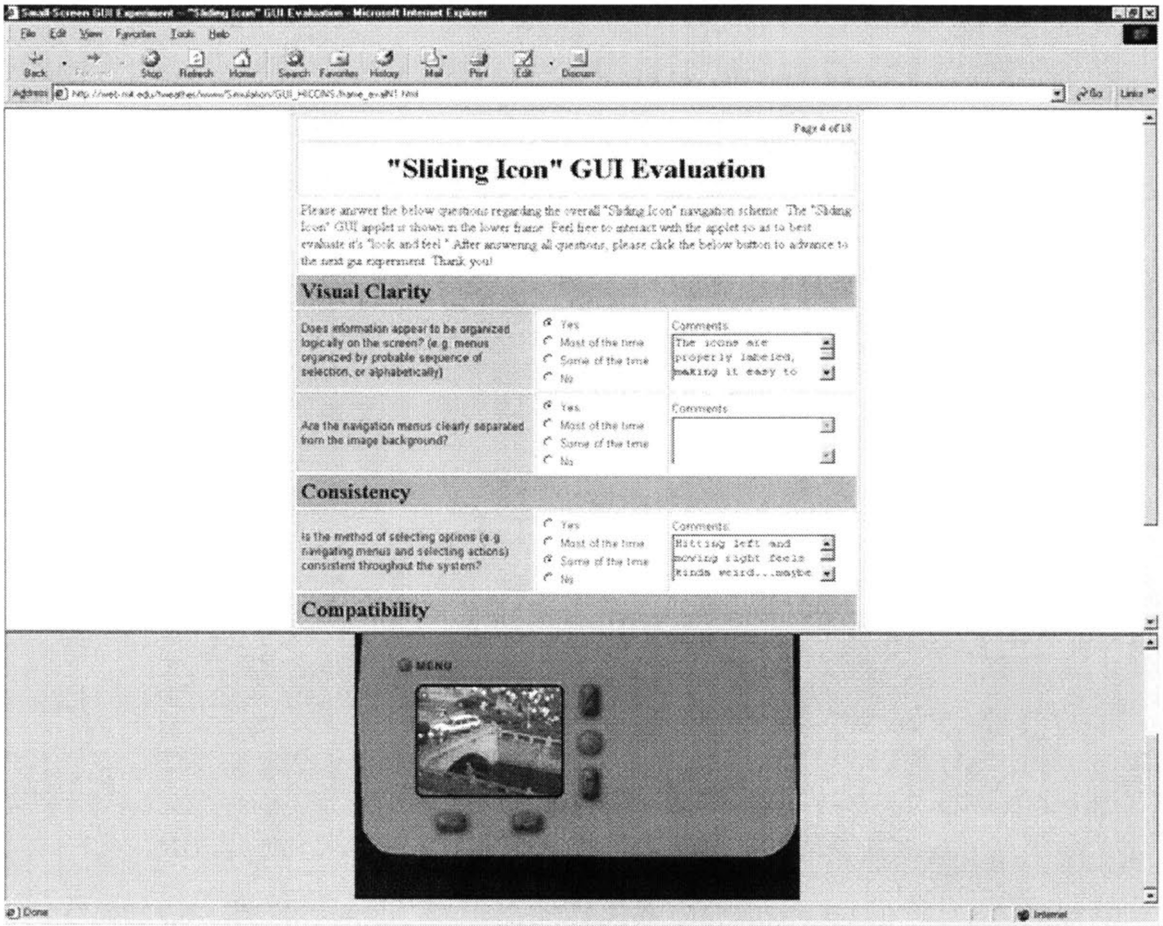


Figure B-7 Prototype Evaluation Page for the Sliding-Icon scheme

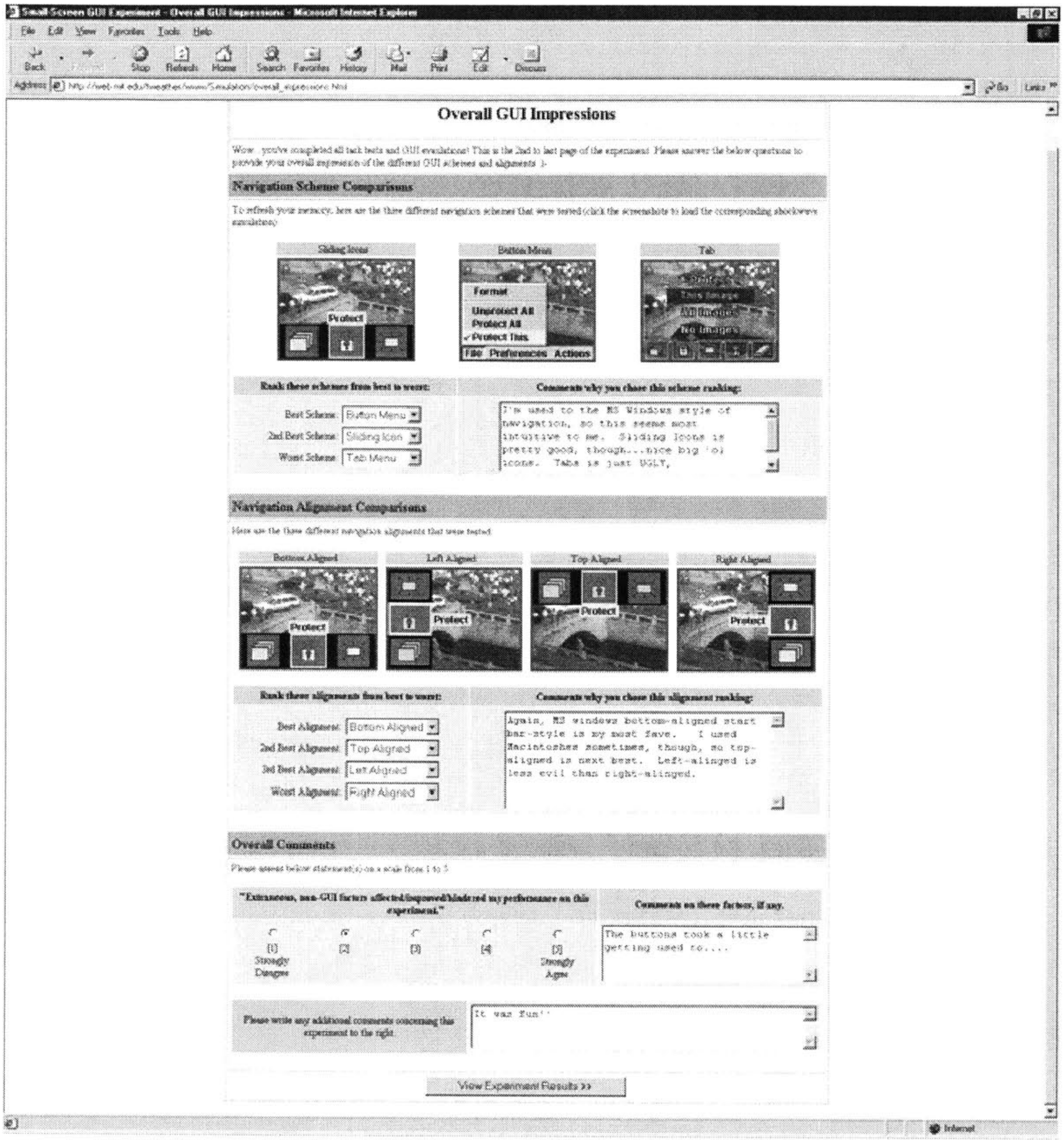


Figure B-8 Overall Impressions Page

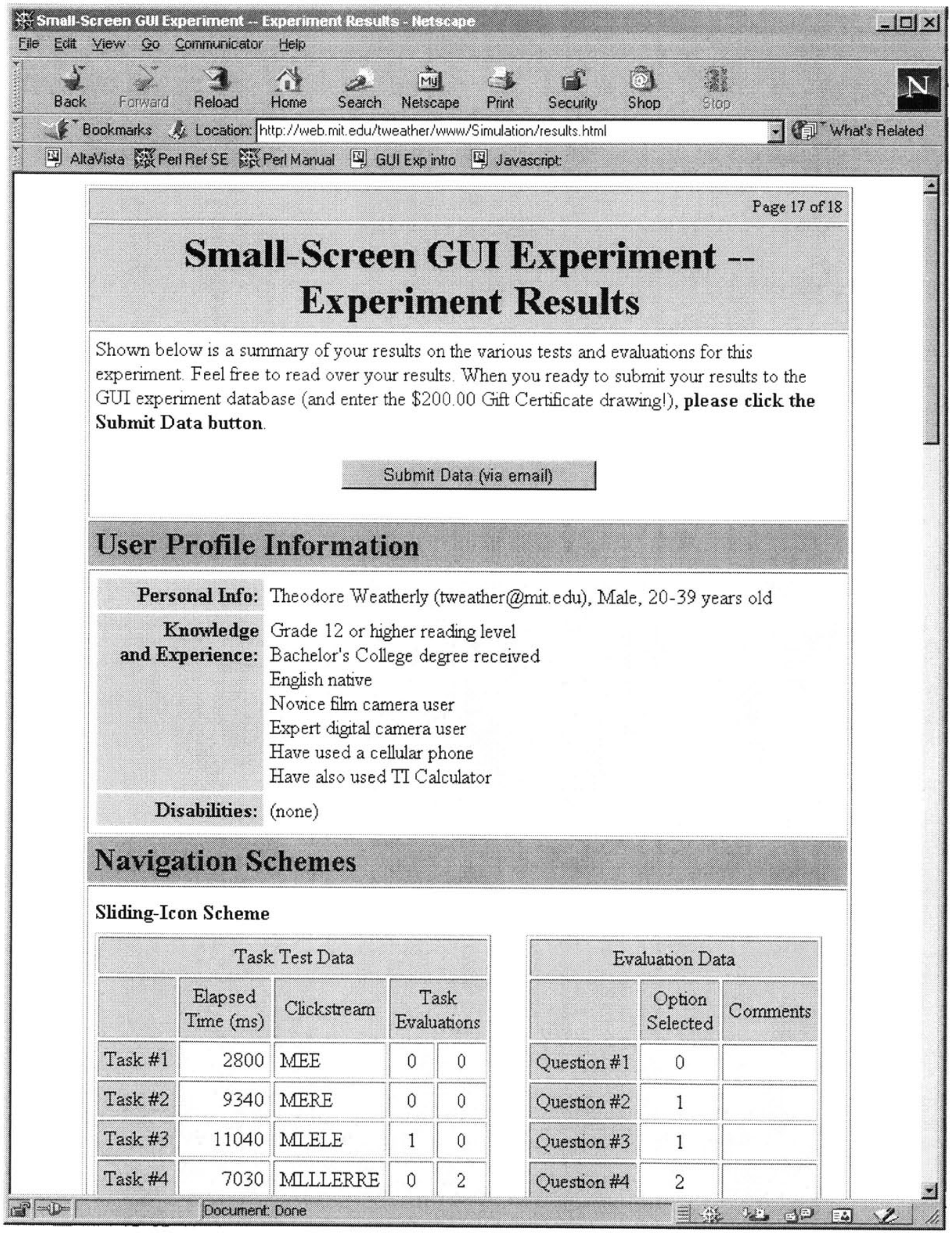


Figure B-9 Results Summary and Submission Page

Appendix C: Question-Based Attitude Assessment Graphs and Tables

This appendix contains graphs and tables pertaining to the attitude assessment questions answered for the hyperactive usability experiment.

C.1 Navigation Scheme Attitude Assessments

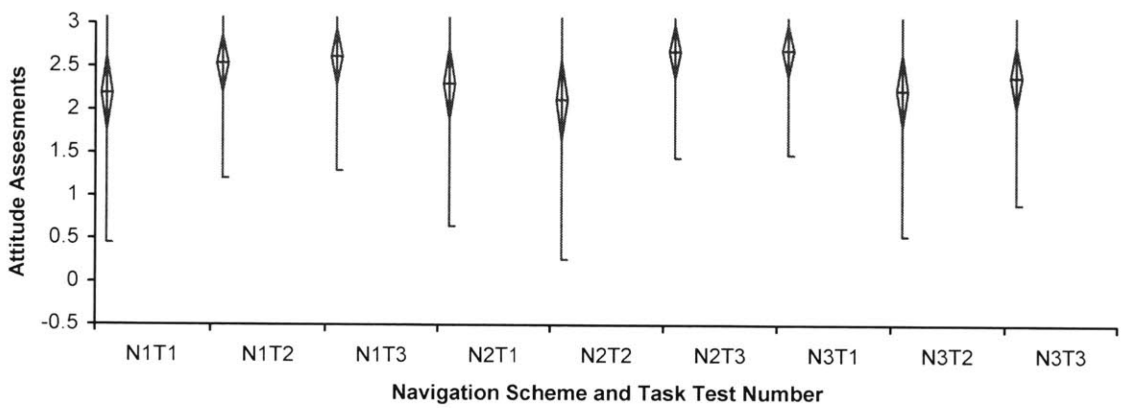


Figure C.1-1 Graphical Display of Assessments for "Does the sequence of activities required to complete this past task follow what you would expect?"

Task ID	Mean Ranking (0-3)	StDev
N1T1	2.19	1.06
N1T2	2.54	0.81
N1T3	2.62	0.80
N2T1	2.31	1.01
N2T2	2.12	1.13
N2T3	2.68	0.75
N3T1	2.69	0.74
N3T2	2.23	1.03
N3T3	2.38	0.90

Table C.1-1 Assessment Data for "Does the sequence of activities required to complete this past task follow what you would expect?"

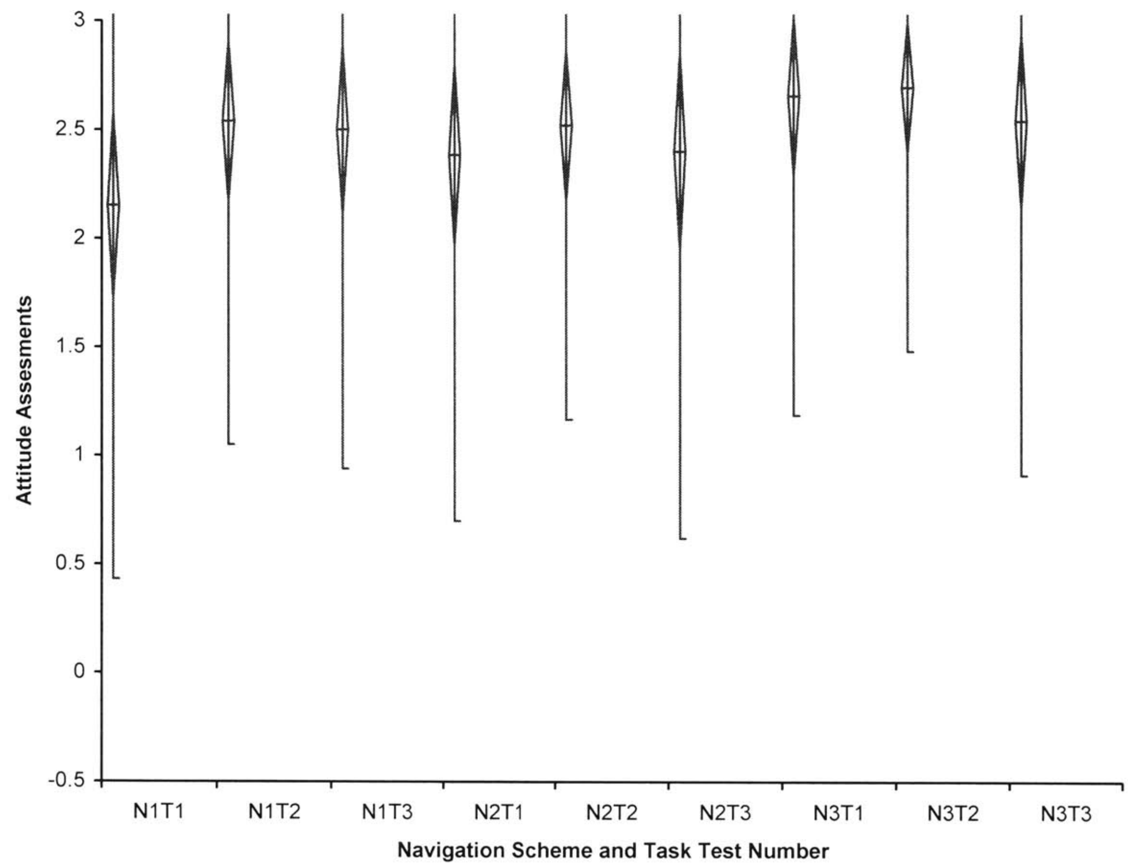


Figure C.1-2 Graphical Display of Assessments for "Is it clear what stage the camera has reached during your navigation for this past task?"

Task ID	Mean Ranking (0-3)	StDev
N1T1	2.15	1.05
N1T2	2.54	0.90
N1T3	2.50	0.95
N2T1	2.38	1.02
N2T2	2.52	0.82
N2T3	2.40	1.08
N3T1	2.65	0.89
N3T2	2.69	0.74
N3T3	2.54	0.99

Table C.1-2 Assessment Data for "Is it clear what stage the camera has reached during your navigation for this past task"

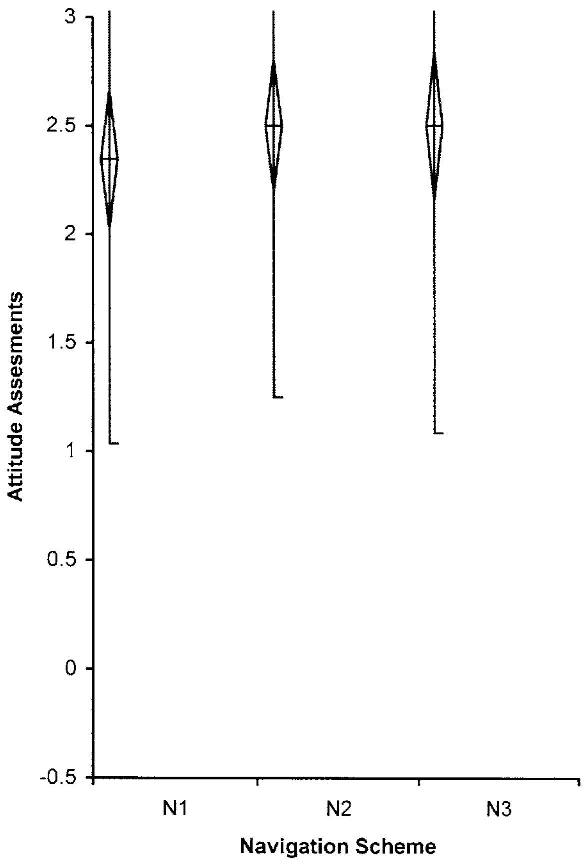


Figure C.1-3 Graphical Display of Assessments for "Does information appear to be organized logically on the screen?"

Task ID	Mean Ranking (0-3)	StDev
N1	2.35	0.80
N2	2.50	0.76
N3	2.50	0.86

Table C.1-3 Assessment Data for "Does information appear to be organized logically on the screen?"

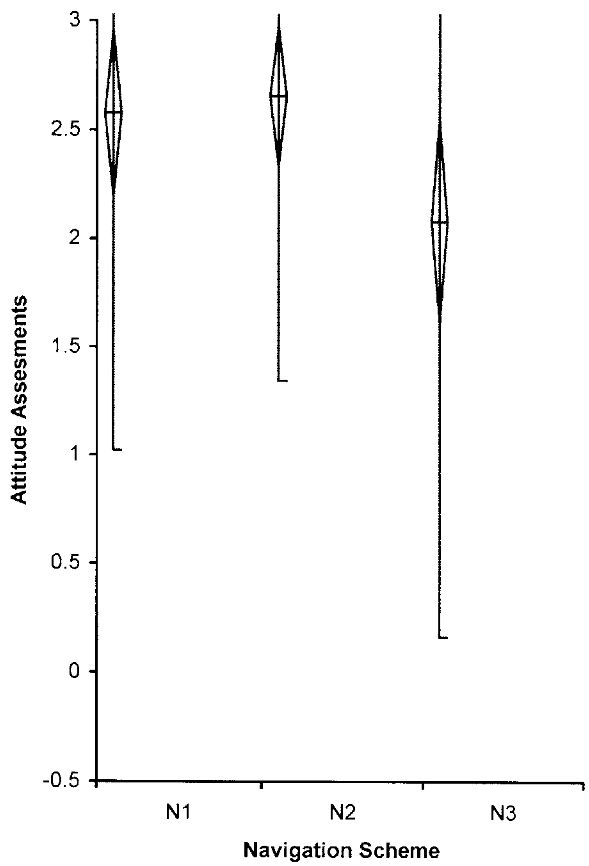


Figure C.1-4 Graphical Display of Assessments for "Are the navigation menus clearly separated from the image background?"

Task ID	Mean Ranking (0-3)	StDev
N1	2.58	0.95
N2	2.65	0.80
N3	2.08	1.16

Table C.1-4 Assessment Data for "Are the navigation menus clearly separated from the image background?"

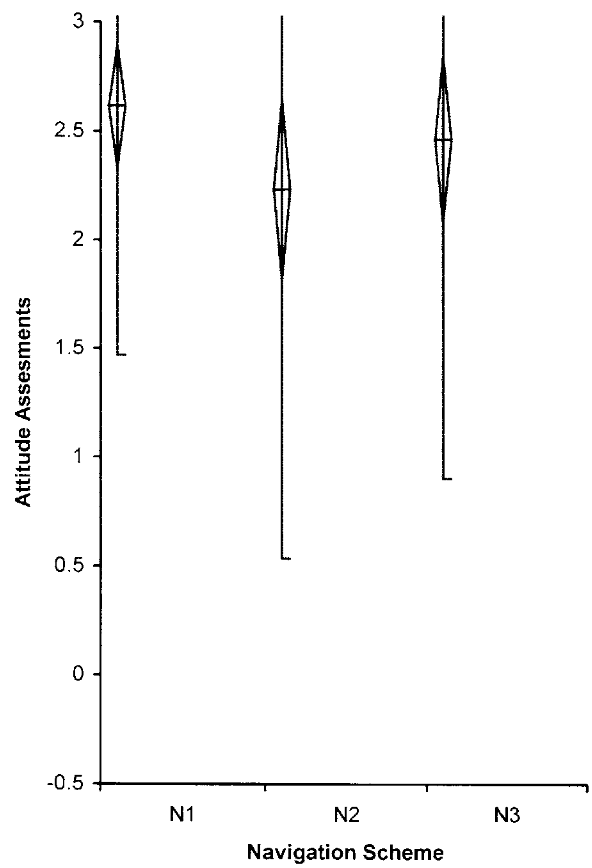


Figure C.1-5 Graphical Display of Assessments for "Is the method of selecting options consistent throughout the system?"

Task ID	Mean Ranking (0-3)	StDev
N1	2.62	0.70
N2	2.23	1.03
N3	2.46	0.95

Table C.1-5 Assessment Data for "Is the method of selecting options consistent throughout the system?"

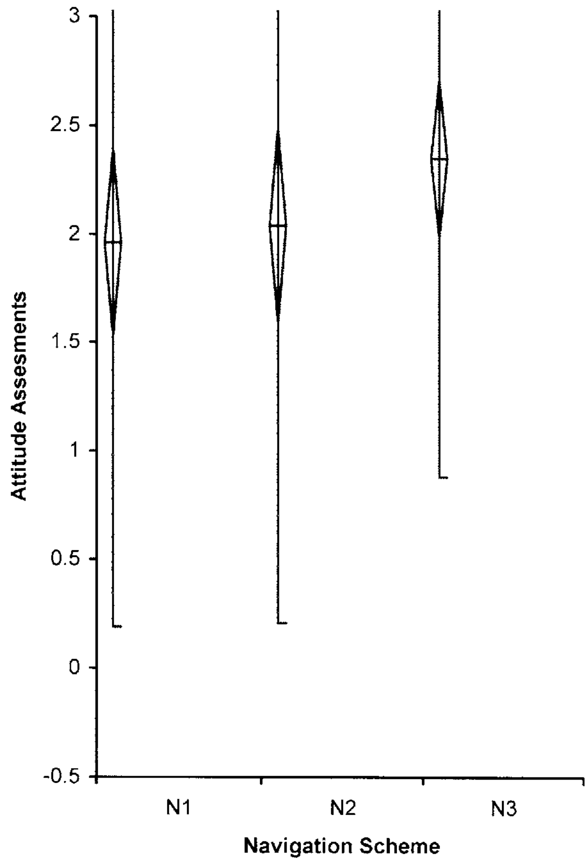


Figure C.1-6 Graphical Display of Assessments for "Does navigation using this navigation style feel natural?"

Task ID	Mean Ranking (0-3)	StDev
N1	1.96	1.08
N2	2.04	1.11
N3	2.35	0.89

Table C.1-6 Assessment Data for "Does navigation using this navigation style feel natural?"

C.2 Navigation Alignment Attitude Assessments

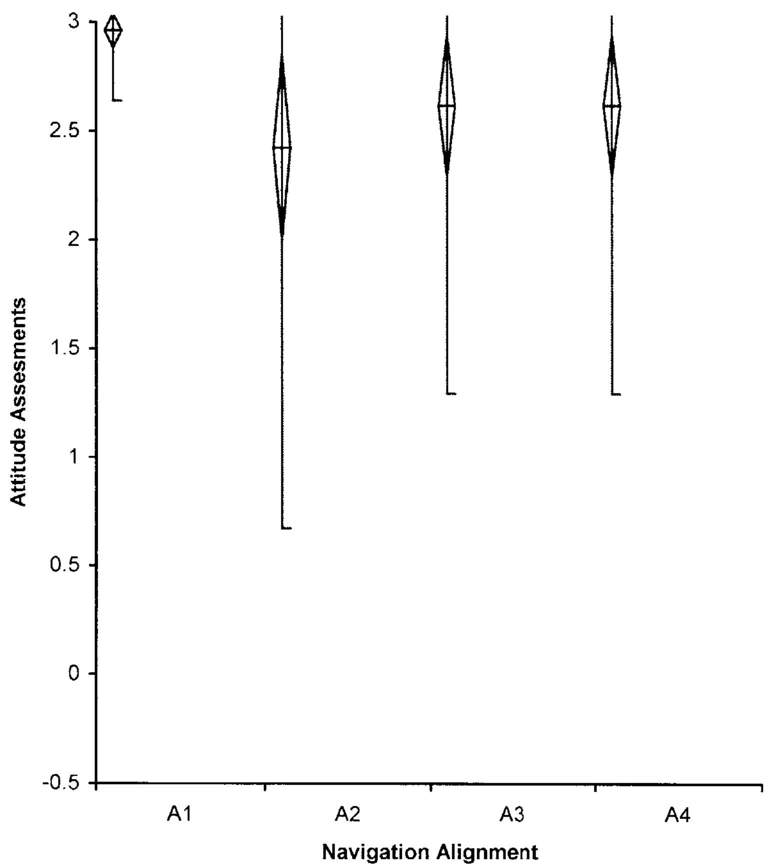


Figure C.2-1 Graphical Display of Assessments for "Are the menu options aligned properly on the screen?"

Task ID	Mean Ranking (0-3)	StDev
A1	2.96	0.20
A2	2.42	1.06
A3	2.62	0.80
A4	2.62	0.80

Table C.2-1 Assessment Data for "Are the menu options aligned properly on the screen?"

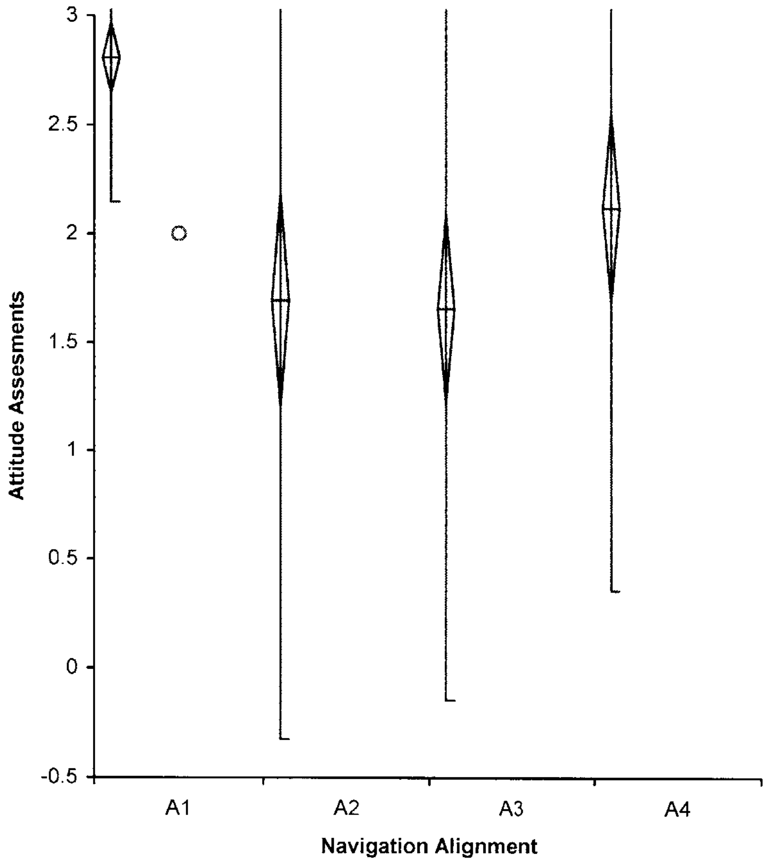


Figure C.2-2 Graphical Display of Assessments for "Does the alignment of the navigation menus feel natural?"

Task ID	Mean Ranking (0-3)	StDev
A1	2.81	0.40
A2	1.69	1.23
A3	1.65	1.09
A4	2.12	1.07

Table C.2-2 Assessment Data for "Does the alignment of the navigation menus feel natural?"

References

[1] Zoom.net. (2000) "ADSL – The Market". [Online] URL: http://www.zoom.net/adslmkt.htm (Accessed: March 29, 2000).

[2] Sullivan, K. (1996) "The Windows® 95 User Interface: A Case Study in Usability Engineering". Proceedings from *CHI '96 Human Factors in Computing Systems*, pg. 473-480.

[3] Hartson, H.R., Castillo, J.C., Kelso, J., Kamler, J., and Neale, W.C. (1996) "Remote Evaluation: The Network as an Extension of the Usability Laboratory". Proceedings of *CHI'96 Human Factors in Computing Systems*, pg. 228-235.

[4] Kashambuzi, M.M. (1999) "Design, Data Mining, and Analysis of Hyperactive Environments". Master of Science Thesis, MIT Department of Mechanical Engineering.

[5] Preece, J. (1993) *A Guide To Usability*. Addison-Wesley.

[6] Shackel, B. (1991b) "Human factors for informatics usability – background and overview", in *Human Factors for Informatics Usability* (eds B. Shackel and S. Richardson). Cambridge University Press, Cambridge.

[7] Lingaard, G. (1994) *Usability testing and system evaluation: A guide for designing useful computer systems*. Chapman & Hall.

[8] Jordan, Thomas, Weerdmeester and McClelland (1996) *Usability Evaluation in Industry*. Taylor & Francis.

[9] Ravden, S. Johnson, G. (1989) *Evaluating Usability of Human-Computer Interfaces: A Practical Method*. Ellis Horwood Limited, Chinchester.

[10] Adriaans, P., and Zantinge, D. (1996) *Data Mining*. Harlow, England. Addison-Wesley Publishing Co.

[11] Bailey, R. W. (1989) *Human Performance Engineering*. Prentice Hall.

Bibliography

Epstein, B.A., (1998) *Lingo In A Nutshell: A Desktop Quick Reference.* O'Reilly & Associates, Inc.

Flanagan, D. (1998) *Javascript: The Definitive Guide.* O'Reilly & Associates, Inc.

Hopper, S., Hambrose, H., Kanevsky, P. (1996) "Real World Design in the Corporate Environment: Designing an Interface for the Technically Challenged". Proceeding of *CHI '96 Human Factors in Computing Systems*, pg. 489-495.

Kamba et Al. (1996) "Using Small Screen Space More Efficiently". Proceedings of *CHI'96 Human Factors in Computing Systems*, pg. 383-390.

Liu, H., Motoda, H. (1998) *Feature Selection For Knowledge Discovery and Data Mining.* Kluwer Academic Publishers, Boston.

Macleod, M., Rosemary, B., Nigel, B. (1997) "The MUSiC Performance Measurement Method". *Behaviour and Information Technology* Volume 16. [Online] URL: http://www.usability.serco.com/nonframe/publs.html (Last Accessed: August 28, 2000).

Noyes, J.M., Cook, M. (1999) *Interface Technology – The Leading Edge.* Research Studies Press Ltd., Baldock, Hertfordshire, England.

Page, S.R., Johnsgard, T.J., Albert, U., Allen, C.D. (1996) "User Customization of a Word Processor". Proceedings of *CHI'96 Human Factors in Computing Systems*, pg. 340-346.

Rosenzweig, G. (1999) *Special Edition Using Macromedia Director 7.* Que Corporation, Indianapolis, Indiana.

Sonwalker, N., Kovgan, A., Wiggins, R.J. (1999) "A Comparative Evaluation of Distance Learning Modalities: Desktop Learning in the 21st Century and Beyond". Proceedings of *Online Educational Conference*, Berlin.

Stephens, R.K., Plew, R.R. (1999) *Sams Teach Yourself SQL in 21 Days, Third Edition.* Sams Publishing, Indianapolis.