Usability of Navigation Tools in Software for Browsing Genetic Sequences

A thesis

submitted in partial fulfilment

of the requirements for the Degree of

Master of Applied Computing

at

Lincoln University

by

Paul Rutherford

Lincoln University

2008

[This page is intentionally blank]

Abstract of a thesis submitted in partial fulfilment of the requirements for the Degree of Master of Applied Computing

Usability of Navigation Tools in Software for Browsing Genetic Sequences

By Paul Rutherford

Software to display and analyse DNA sequences is a crucial tool for bioinformatics research. The data of a DNA sequence has a relatively simple format but the length and sheer volume of data can create difficulties in navigation while maintaining overall context. This is one reason that current bioinformatics applications can be difficult to use.

This research examines techniques for navigating through large single DNA sequences and their annotations. Navigation in DNA sequences is considered here in terms of the navigational activities: exploration, wayfinding and identifying objects. A process incorporating user-centred design was used to create prototypes involving panning and zooming of DNA sequences. This approach included a questionnaire to define the target users and their goals, an examination of existing bioinformatics applications to identify navigation designs, a heuristic evaluation of those designs, and a usability study of prototypes.

Three designs for panning and five designs for zooming were selected for development. During usability testing, users were asked to perform common navigational activities using each of the designs. The "Connected View" design was found to be the most usable for panning while the "Zoom Slider" design was best for zooming and most useful zooming tool for tasks involving browsing. For some tasks the ability to zoom was unnecessary.

The research provides important insights into the expectations that researchers have of bioinformatics applications and suitable methods for designing for that audience. The outcomes of this type of research can be used to help improve bioinformatics applications so that they will be truly usable by researchers.

Keywords: bioinformatics, deoxyribonucleic acid, usability, user interface, navigation, browsing, panning, zooming, user-centred design

[This page is intentionally blank]

Acknowledgements

I have a great number of people to thank for supporting me throughout this research.

Firstly, my thanks to those who had the confidence, passion and enthusiasm that led me into this research in the first place: John McCallum of Crop and Food Research and Clare Churcher of the Applied Computing Group, Lincoln University. Without either of you this research could not have happened.

To those who supervised me through the writing process, Walt Abell and Alan McKinnon, both of the Applied Computing Group, Lincoln University; keeping an eye on me and keeping me working.

To my fellow colleagues in the Applied Computing Group, particularly Flo Weingartner, Shirley Gibbs, Sharon Long, Theresa McLennan and Keith Unsworth, and my fellow postgraduates.

To my family both biological and extended who have cared for me, especially my mum, Helen Thornton, and my god-parents, John and Helen Webster.

To my friends who have kept me happy and relatively sane through out all the years this has taken place, particularly Cheri Harema, Haidee Pyne, Adam Smith, Rhea Ramadurai, Gabriele Benaglia, James Maxwell and Ivy Harper. There are many people I have neglected over this time and I look forward to catching up with them soon.

To those who have supported me at Lincoln University, including Tracey Shields, Chrissy Gibson, Jane Swift, Caitriona Cameron, Glenda Hicks and Bernadette Mani.

To Student Health and Support who have kept me healthy and well, particularly Dr Hamish Dunn, Doreen Lassen, Jeanie Sanford, Fiona Gould, Jenny Rose and Rachel Higgens.

And all the other friendly, kind and caring people in my life. You make a difficult journey manageable and more pleasant.

This project had the involvement of Crop and Food Research, and was funded by the Foundations for Research, Science and Technology (FRST) (Contract no. C02X0203)

[This page is intentionally blank]

Table of Contents

Chapter 1: Introduction ... 1

Chapter 2: Literature Review ... 3

2.1 Genetic Sequences ... 3

2.1.1 DNA Sequences ... 3

2.1.2 Bioinformatics ... 5

2.2 User Interfaces ... 5

2.2.1 User Interface Approaches ... 6

2.2.1.1 Direct Manipulation ... 6

2.2.1.2 Components for Interaction ... 7

2.2.2 Prototypes ... 9

2.3 Navigation ... 10

2.3.1 Information Spaces ... 11

2.3.2 Visualising an Information Space ... 11

2.3.3 Navigational Strategies ... 13

2.3.4 Navigational Activities ... 15

2.3.5 Assisting Navigation ... 16

2.4 Usability ... 17

2.4.1 Attributes of Usability ... 18

2.4.2 Principles for Usability ... 18

2.4.3 User-Centred Design ... 20

2.4.3.1 User-Centred Design Activities ... 21

2.5 Usability Evaluation ... 21

2.5.1 Methods of Usability Evaluation ... 22

2.5.2 Usability Inquiry ... 24

2.5.3 Usability Inspection ... 24

2.5.4 Usability Testing 25
2.5.4.1 People Involved in a Usability Trial 26
2.5.4.2 Protocol of Usability Testing 26
2.5.4.3 Data Collection 27
2.5.5 Measures of Usability 28
2.5.5.1 Objective Measures 28
2.5.5.2 Subjective Measures 29
2.5.5.3 Severity of Usability Problems 30
2.6 Application to Bioinformatics 30
2.6.1 User Interface 30
2.6.2 Navigation 31
2.6.3 Usability 32
2.7 Summary 33
Chapter 3: Proposed Research 34
3.1 Target Users 34
3.2 Research Aim and Objectives 34
3.3 Scope 35
3.4 Summary 36
Chapter 4: Defining the Target Users 37
4.1 Content of the Questionnaire 37
4.1.1 Demographics 37
4.1.2 Tools 38
4.1.3 Tasks 38
4.2 Findings from the Questionnaire 39
4.2.1 Demographics 39
4.2.2 Tools 41
4.2.3 Tasks 45

4.3 Conclusions 46

Chapter 5: Examination of Current Applications 47

5.1 Applications Examined 47

5.1.1 APIC (Bisson and Garreau, 1995) 47

5.1.2 Apollo (Lewis et al., 2002) 48

5.1.3 Artemis (Rutherford et al., 2000) 48

5.1.4 BLAST (McGinnis & Madden, 2004) 49

5.1.5 ChARMView (Myers, Chen, & Troyanskaya, 2005) 49

5.1.6 DNAMAN (Woffelman, 2004) 50

5.1.7 Ensembl (Hubbard et al., 2002) 50

5.1.8 GeneViTo (Vernikos et al., 2003) 51

5.1.9 GAP (Bonfield, Smith, & Staden, 1995) 51

5.1.10 Genotator Browser (Harris, 1997) 52

5.1.11 Gestalt (Glusman & Lancet, 2000) 52

5.1.12 MEGA (Kumar, Nei, Dudley, & Tamura, 2008) 52

5.1.13 NCBI Map Viewer (Wheeler et al., 2005) 53

5.1.14 NEBcutter (Vincze, Posfai, & Roberts, 2003) 53

5.1.15 Primer3 WWW Interface (Rozen & Skaletsky, 2000) 54

5.1.16 RegulonDB (Salgado et al., 2001) 54

5.1.17 SeqScape (Applied Biosystems, 2004) 54

5.1.18 Sequencher (Gene Codes Corporation, 2003) 55

5.1.19 SeqVista (Hu et al., 2003) 55

5.1.20 UCSC Browser (Karolchik et al., 2002) 56

5.2 Summary of Techniques 56

5.3 Other Applications 57

5.3.1 Adobe Acrobat Reader (Adobe, 2004) 58

5.3.2 ECan Online GIS (ECan, 2005) 58

5.3.3 Google Maps (Google, 2005) 58

5.4 Conclusion 59

Chapter 6: Heuristic Evaluation 60

6.1 Evaluation Process 60

6.1.1 Context 60

6.1.2 Method 60

6.2 Evaluation of Designs for Panning 61

6.3 Evaluation of Designs for Zooming 63

6.4 Discussion 67

6.5 Conclusion 68

Chapter 7: Prototypes 69

7.1 Visualisation of Genetic Sequences 69

7.1.1 Creating the Sequence 71

7.1.2 Panning the Sequence 71

7.1.3 Zooming the Sequence 72

7.2 Prototypes for Panning 74

7.2.1 Panning Buttons 74

7.2.2 Scroll Bar 75

7.2.3 Connected View 76

7.3 Prototypes for Zooming 77

7.3.1 Zoom Buttons 77

7.3.2 Zoom Slider 77

7.3.3 On-view Slider 78

7.4 Other Prototypes 79

7.4.1 Dynamic Zoom 79

7.4.2 Magnifying Glass Tool 81

Chapter 8: Usability Study Method 82

8.1 Trial Organisation 82

8.2 Trial Process 84

8.2.1 Section One: Terminology 85

8.2.2 Section Two: Usability Test with Designs for Panning 87

8.2.3 Section Three: Symbol and Orientation Meanings 89

8.2.4 Section Four: Usability Test with Designs for Zooming 92

8.2.5 Section Five: Brief Investigation of Other Interfaces 93

8.3 Data Collection 93

8.4 Pre-testing 94

8.5 Analysis 95

Chapter 9: Results and Discussion 97

9.1 Data Collection 97

9.2 Section One: Terminology 97

9.3 Section Two: Usability Test with Designs for Panning 100

9.3.1 Performance Measures 100

9.3.2 Panning Buttons Design 102

9.3.3 Scroll Bar Design 104

9.3.4 Connected View Design 105

9.3.5 Participant Preferences 107

9.3.6 Other Discussion 107

9.3.7 Conclusion 108

9.4 Section Three: Symbol and Orientation Meanings 108

9.5 Section Four: Usability Test with Designs for Zooming 110

9.5.1 Performance Measures 110

9.5.2 Zoom Buttons Design 113

9.5.3 Zoom Slider Design 114

9.5.4 On-view Slider Design 115

9.5.5 Participant Preferences 116

9.5.6 Other Discussion 117

9.5.7 Conclusion 118

9.6 Section Five: Brief Investigation of Other Interfaces 118

9.6.1 Dynamic Zoom 119

9.6.2 Magnifying Glass Tool 119

9.7 Participant Feedback 120

9.7.1 Visualisation 120

9.7.2 Tasks 120

9.7.3 Sequences 121

Chapter 10: Conclusions 122

10.1 Summary of Results 122

10.2 Limitations 124

10.3 Future Work 124

10.4 Final Comments 125

References 127

List of Tables

Table 1 – Responses to Question 1, 'Job Description' 39

Table 2 – Responses to Question 2, 'Computer Use' 39

Table 3 – Responses to Question 3, 'Computer Experience' 40

Table 4 – Responses to Question 4, 'Work with genetic sequence data' 40

Table 5 – Responses to Question 5, 'Computer use for bioinformatics tasks' 41

Table 6 – Responses to Question 6, 'Work on paper for bioinformatics tasks' 41

Table 7 – Responses to Question 8, 'Storing genetic sequences' 42

Table 8 – Responses to Question 9, 'Storing results from bioinformatics applications' 43

Table 9 – Responses to Question 10, 'Use of bioinformatics applications' 43

Table 10 – Responses to Question 11, 'Ability with bioinformatics applications' 44

Table 11 – Tally of instances of methods for providing panning 56

Table 12 – Tally of instances of methods for providing zooming 57

Table 13 – Ranking of panning designs 67

Table 14 – Ranking of zooming designs 67

Table 15 – Showing the action performed by each of the buttons 75

Table 16 – Description of tasks and examples of actual questions 83

Table 17 – The symbol sets used in the study 89

Table 18 – Average number of clicks to complete each task 101

Table 19 – Average time (seconds) to complete each task 101

Table 20 – Number of participant's preferring each design for each task and overall...... 107

Table 21 – The symbol sets as presented to participants 108

Table 22 – The symbol sets as ranked by participants 109

Table 23 – Percentage of participants that used zooming for each design and each task . 110

Table 24 – Average time (seconds) to complete each task 111

Table 25 – Average time to complete each task, for participants who panned only......... 112

Table 26 – Average time to complete each task, for participants who used zooming 112

Table 27 – Number of participant's preferring each design for each task and overall...... 116

List of Figures

Figure 1 – A DNA segment that is 619 bases long displayed with spaces every 10 bases ... 3

Figure 2 – A representation of a sequence showing the blocks of features for a protein 4

Figure 3 – Button component 8

Figure 4 – Scroll bar component with thumb at mid-range.. 8

Figure 5 – Slider component with track and thumb 9

Figure 6 – An example visualisation, which is larger than the screen 12

Figure 7 – An example visualisation at different levels of magnification........................... 12

Figure 8 – Example of the Detail-Only navigational strategy 13

Figure 9 – Example of the Pan + Zoom navigational strategy 13

Figure 10 – Example of the Overview + Detail navigational strategy 14

Figure 11 – Example of the Focus + Context navigational strategy 15

Figure 12 – A simplified example of Lorraine and Helt's visualisation scheme 31

Figure 13 – A sample sequence overview 32

Figure 14 – Mouse tools for selecting level of detail in ChARMView.............................. 49

Figure 15 – Control for selecting level of magnification in Ensembl 50

Figure 16 – Circular map for selecting location in the sequence in GeneViTo 51

Figure 17 – Control for selecting level of magnification in NCBI Map Viewer................. 53

Figure 18 – Mouse tools for selecting level of detail in ECan GIS Viewer 58

Figure 19 – Navigation controls for Google Maps.. 59

Figure 20 – A screen shot from Artemis used to produce the detail view 70

Figure 21 – A screen shot from Artemis used to produce the overview 70

Figure 23 – Showing how panning was performed... 72

Figure 24 – Showing two different levels of magnification for the same region 73

Figure 25 – Showing how compressing and stretching was used to create smooth transitions for magnification 73

Figure 26 – The Panning Buttons prototype 74

Figure 27 – The Scroll Bar prototype 76

Figure 28 – The Connected View prototype 76

Figure 29 – The Zoom Buttons prototype 77

Figure 30 – The Zoom Slider prototype 78

Figure 31 – Showing the behaviour of the On-View Slider 79

Figure 32 – The Dynamic Zoom tool buffer zone 80

Figure 33 – Showing the effect of different actions using dynamic zooming 80

Figure 34 – Buttons provided to select mode 81

Figure 35 – Mouse cursor symbols used to portray mode 81

Figure 36 – Screen showing trial progress 85

Figure 37 – An example of how concepts were presented to participants. 86

Figure 38 – An example of how the designs were presented to participants 87

Figure 39 – Showing how tasks were presented to participants 88

Figure 40 – Showing how the designs were presented at the end of the section 88

Figure 41 – Example screen for investigating meaning of symbols 90

Figure 42 – Showing the screen for investigating orientation of buttons for zooming 90

Figure 43 – Example of how button sets were presented to participants 91

Figure 44 – Example of how button sets were presented to participants once selected 91

Figure 45 – Showing how the designs were presented at the end of the section 92

Figure 46 – Showing how the Magnifying Glass Tool prototype was presented 93

Figure 47 – Four of the items demonstrated to participants 98

Figure 48 – Two of the terms demonstrated to participants, circled. 99

Figure 49 – Panning buttons as presented in the usability study 102

Figure 50 – Chart showing typical behaviour of a participant using the Panning Buttons 103

Figure 51 – Scroll bar as presented in the usability study 104

Figure 52 – Chart showing typical behaviour of a participant using the Scroll Bar 104

Figure 53 – Connected view as presented in the usability study 105

Figure 54 – Zooming buttons as presented in the usability study 113

Figure 55 – Zoom Slider as presented in the usability study 114

Figure 56 – On-view Slider as presented in the usability study 115

Figure 57 – Showing typical behaviour for the ‘Find Feature’ task 117

Figure 58 – Buttons provided to select mode 119

Figure 59 – Mouse cursor symbols used to portray mode 119

[This page is intentionally blank]

Chapter 1: Introduction

Biology is the study of living organisms, “from the structure and function of simple molecules to the interactions among the hundreds or thousands of different types of organisms living together in a particular region” (Purves, 1995). Each cell contains the genetic sequence, deoxyribonucleic acid (DNA), which is the genetic template for the organism. The analysis of DNA is central to bioinformatics and software is commonly used to work with genetic sequences on computers.

The technology to determine the base sequence of a DNA strand was developed in the mid-1970s and the volume of such data has been growing exponentially since then. Bioinformatics tools, where computing and statistical techniques are applied to such data, are now commonly used. This relationship between computers and research users necessitates an interface by which the computer can display the data and the user can interact with it. Research users typically have little formal training in information technology and software developers often have little understanding of the research needs. This has led to difficulties because bioinformatics user interfaces have not been designed with the research users in mind. In addition, when applications are designed by research users themselves, they usually focus on the required functionality with little consideration of the principles of usability. Research has shown that users experience a steep learning curve for existing bioinformatics applications and that users can be overwhelmed by the complexity of performing standard tasks (Javahery, Seffah et al. 2004).

This is compounded by the technical difficulties of dealing with potentially long and large volumes of data. With the volume of data in typical DNA sequences it is not possible to display it all at one time. Solutions for this mean that the user can either see the detail for a part of the sequence or all the sequence but with some detail omitted. In navigating this, users can easily lose context.

This research will apply knowledge from the fields of user interface design, navigation and usability to bioinformatics applications. How DNA sequence information may be interacted with will be examined, with the aim of developing and evaluating ways of displaying and navigating genetic sequences on computers for use by bioinformatics researchers.

The literature review in Chapter Two discusses background knowledge feeding into this research, including genetics, user interfaces, navigation and usability. The context of the proposed research is set in Chapter Three, defining the target user group, the research aims and objectives and the scope of the research.

Chapters Four to Nine contain the body of research performed, the process of which includes: a questionnaire to define the target users and their goals (Chapter Four), an examination of existing bioinformatics applications to identify designs for navigation (Chapter Five), a heuristic evaluation of those designs (Chapter Six) and a usability study. The method of the usability study will be described in Chapter Eight and results and discussion in Chapter Nine.

Chapter Ten is the conclusion for this research. It covers the main results found through the study, limitations of the study and proposes future work. The appendices contain each of the forms used in the research, including the questionnaire form, usability study script, usability study participant preferences form and usability study observers sheet.

Chapter 2: Literature Review

2.1 Genetic Sequences

Biology is the study of living organisms and these living organisms are composed of units called cells. Inside each cell is, essentially, an instruction manual: the genetic sequence. Genetics is a topic within biology that examines the genetic sequence, how that information is shared from parent to offspring, and how this information changes over time. The uses of genetic sequences include the ability to identify species and individuals, to investigate inherited diseases and to manipulate organisms by genetic engineering. The primary sequence type is deoxyribonucleic acid (DNA).

2.1.1 DNA Sequences

DNA is an information-coding molecule that is in every cell of every living organism and it tells the cell what it needs to produce and how to behave. DNA has a relatively simple structure, composed of regular units that link together to form a chain. These units, called 'nucleotides' or 'bases', come in four forms, the chemicals adenine, cytosine, guanine and thymine. For brevity these are referred to as A, C, G, and T. A DNA sequence can be represented by a sequence of those four units or letters (see Figure 1). These sequences are often very long, for example the entire human DNA sequence, the human genome, is 3 million bases long.

```
1   ttaggataac tggttgtcct aatggctgtg ccagacctta catggctgag ctagggtttg
61  ttggtgatgg cccaaatagc taccagatat ggcttggagg aacacctaac cagagtacac
121 tagcgaagtg ttttatgaat aaggtgaaga tccaaaaatt tgagagcgtt ttagaaccgc
181 tgttcaatga ctggaaagta aatcgcaaga cagaagaatc atttggtgaa tttacaaacc
241 gaattggctt tgaaaagctg cttgaggttg tagagcagtg ggatagttct aaaaattaat
301 ctgttccccc tttaccacga gggttttgaa taagtgcgtg aagaataata atataagaag
361 ttggatggtt caggtaatcg attaatccat tcttggacca ctgagacagc gaggacccat
421 ttattatgtc ctaaatatgc aacgagtgca gcgtgtaatg tattttgata gttccgtttt
481 cagtttgtat ctttcgtgtg tatgttcttg ttttcttcgg tgaaaactaa gaatatgaat
541 ggtttatgct tacattttgt tcaagttcgg aataaaaatc atgaattcat ccattaccaa
601 taaaaaaaaa aaaaaaaaa//
```

Figure 1 – A DNA segment that is 619 bases long displayed with spaces every 10 bases

The information from DNA sequences is directly translated into a more complex molecule structure which produces proteins (Cohen, 2004). Proteins are the molecules required for cellular function and structure. The structure of a protein, and therefore the function, can be predicted by understanding the DNA sequence that produces it (Luscombe, Greenbaum & Gerstein, 2001).

DNA contains a number of “features” which are segments of the DNA sequence that are known to have a particular purpose, for example the string of ‘A’s at the end of the sequence shown in Figure 1 is known to occur at the end of a protein.

Features vary in size from one base to thousands of bases long. An example of a single base feature is a single nucleotide polymorphism (SNP). This is one base that may change between different sources of the same sequence. An example of a feature that is thousands of bases long is a gene, which is all of the sequence that is required to produce a particular protein (Loraine & Helt, 2002).

Features may be simple, or they may be part of a larger construct. Some sequences of DNA are known to encode for a particular feature, for example the sequence ‘ATG’ (Adenine-Thymine-Guanine) encodes for the start of a protein. A gene, the sequence that produces that protein, is a larger construct including several features (see Figure 2):

- A Start Codon, which marks the start of the protein sequence
- Exons, which encode for the protein product, collectively called the ‘coding region’
- Introns, which fragment the coding region and are removed in producing the protein and
- A Stop Codon, which marks the end of the protein.

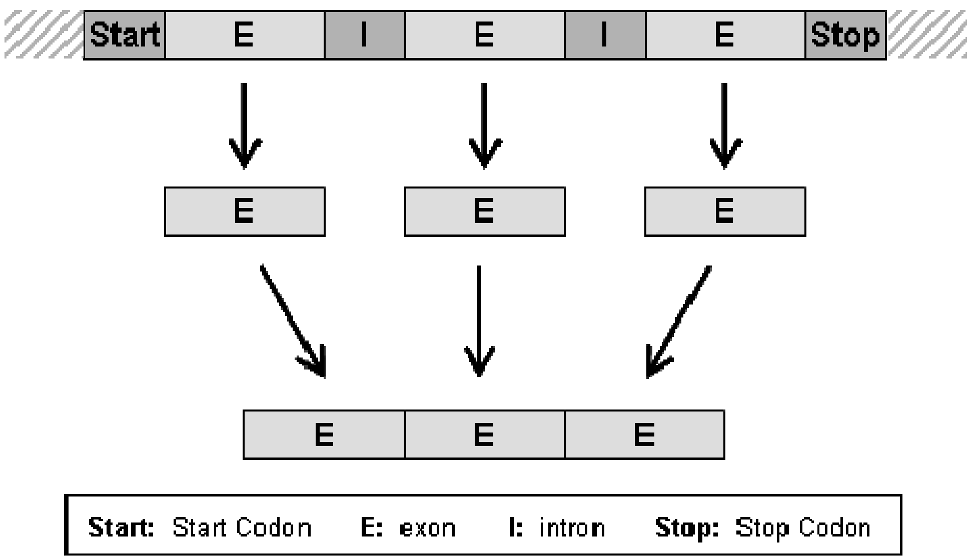


Figure 2 – A representation of a sequence showing the blocks of features for a protein

The cutting of introns from the sequence and joining of exons to produce the protein sequence is illustrated in Figure 2. It is often necessary for biologists to view the sequence data in the context of those features (Loraine & Helt, 2002).

A regular task for biologists is to annotate DNA sequences with information about features of those sequences (Luscombe et al. 2001). The purpose of this is to organise the sequence data and to understand the biological meanings of it (Liu et al., 2003).

2.1.2 Bioinformatics

Biological research is producing massive volumes of data, making computational methods "indispensible" for dealing with them (Luscombe et al., 2001). To give some perspective, in April 2008 there were 89 billion bases and 85 million sequences stored in the National Center for Biotechnology Information (NCBI) GenBank database (National Center for Biotechnology Information, 2008).

As a consequence of the growing volume of biological data, a new field of biology, bioinformatics, has developed. Bioinformatics is an interdisciplinary science that applies computer science and statistical tools to deal with the volumes of biological data (Luscombe et al., 2001) and to recognise, understand and predict their meaning (Cohen, 2005).

DNA sequences may be dealt with on computers as strings of letters representing the bases. Regular tasks with DNA sequences on the computer are to annotate those sequences, to design experiments based on that specific sequence, and to compare sequences with other known and understood sequences to infer meaning. "How to handle these data, make sense of them, and render them accessible to biologists working on a wide variety of problems is the challenge facing bioinformatics" (Roos, 2001).

2.2 User Interfaces

The user interface is the part of the system that the user sees, hears and interacts with, allowing the user and computer to communicate with each other (Tidwell, 2005). Other parts of the system not required for user understanding remain hidden behind the interface (Lauesen, 2005) and "from the point of view of the user, the interface IS the system" (Norman, 1988).

User interfaces that the typical user encounters include those of the web sites that they visit, that of the operating system installed on their computer and for the applications that they use.

User interfaces differ in the systems that they represent, the interactions offered with that system and how that interactivity is afforded. The appropriateness of a user interface depends upon the user, how it is being used and the task being performed (Nahum & Ward, 2001). As the screen space available is limited, its use needs to be judged carefully (Spence, 1999). “In principle, it is easy to make a user interface” (Lauesen, 2005) however it is “still not easy to design *good* interfaces” (Tidwell, 2005).

2.2.1 User Interface Approaches

The type of user interface considered in this research is a graphical user interface for interacting with an object such as a document, image, video or other visualisation. The main focus of such an interface is a component displaying the object, called the view (Guiard, Beaudouin-Lafon et al., 2004). For this reason the view is usually the largest component in the interface and is generally in the centre of the screen to guide users' attentions to it immediately (Tidwell, 2005).

The object may be interacted with using keyboard commands, direct manipulation using the mouse, or indirectly through other components that interact with the object. The latter two methods will now be discussed.

2.2.1.1 Direct Manipulation

The user interface may permit the user to directly manipulate objects using the mouse. Users may interact with an object by clicking, pointing, dragging or performing other physical interactions as programmed by the developer. Sometimes this method of manipulation is apparent, e.g. a button in the user interface usually looks clickable. Sometimes the method of manipulation is not so apparent, e.g. the user may be able to pan an image object by clicking on that object and dragging it to a new location, however this is not inherently evident.

The main direct manipulation techniques using the mouse are:

- **Pointing**
  Placing the mouse cursor over a location on the object to cause an action.
- **Clicking**
  Pointing at a location on the object and clicking the mouse button to cause an action.
- **Dragging**
  Clicking on a location, dragging the mouse to point at another location and releasing the mouse button.

Each technique may be assigned different actions. The typical action from pointing is to display 'tool tips', these are brief messages that appear when the mouse cursor is hovered over a button. An example of an action from clicking is the 'Magnifying Glass Tool' as implemented in Adobe Acrobat Reader and other applications. In this mode, the user clicks directly on the document to enlarge the view centred on that point.

Typical actions achieved by dragging include: panning the view, 'dynamic zooming', or selecting a rectangular region of the view. Panning the view may be achieved by dragging the object the distance and direction to be moved. 'Dynamic zoom', as implemented in Adobe Acrobat Reader, involves clicking on the view and dragging up to zoom in, or dragging down to zoom out (Sahlin, 2003). Selecting a rectangular region, also called the 'Marquee Tool', is achieved by clicking at the top-left corner region and dragging the mouse to the bottom-right corner and releasing the mouse button.

2.2.1.2 Components for Interaction

User interface components for interacting with another object provide a visual representation and concrete metaphor for the control of an object. Users may interact with a component by pointing, clicking, dragging or performing other physical interactions as programmed by the developer (Shneiderman, 1997), and as a result of that interaction some action is effected on an object of interest.

A user interface component is a building block for user interfaces (Robinson & Flores, 1997). When these components are re-used, they generally look the same, behave the same and perform a similar function, making development faster and the interface more recognisable to users.

Three common user interface components are now discussed.

Buttons

A button is designed to look clickable (Figure 3) and clicking a button causes an action to be performed.

Figure 3 – Button component

Buttons have two user interface events, the first when the button is pressed and the second when the button is released. This enables more complex interactions with a button than a simple click would allow, for example while the button is held down the action may be repeated.

A button may have multiple states which are cycled through at each click of the button, e.g. clicking on the button may toggle between an 'on' and 'off' state. Furthermore, a button may be part of a group of related buttons for which special interactions are provided, e.g. mutual exclusivity, where only one button in the group may be 'on'.

Scroll bars

Scroll bars are commonly used for scrolling through a document that is larger than can fit in the screen.

The typical scroll bar (Figure 4) consists of:

- buttons for moving left or moving right,
- a 'track' representing the range of movement, and
- a bar that can be dragged along the track, commonly referred to as the 'thumb' (Apple Computer, 2004).

Figure 4 – Scroll bar component with thumb at mid-range

The size of the thumb represents the size of the view relative to the entire document. The position of the thumb within the track represents the position of the view within the document. Moving the thumb changes the position of the view.

Methods of interacting with the scroll bar are:

- dragging the scroll bar thumb along the track,
- clicking on scroll buttons at either end of the track to move a fixed amount in that direction, and
- clicking within the track, to move the thumb a fixed amount in that direction (standard behaviour is one window-ful) (Apple Computer, 2004).

The movements caused by the interaction occur immediately.

Sliders

Sliders are commonly used for selecting a value from a continuous range.

The typical slider (Figure 5) consists of:

- a 'track' representing the range of movement, and
- a bar that can be dragged along the track, commonly referred to as the 'thumb' (Apple Computer, 2004).

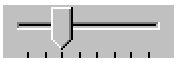


Figure 5 – Slider component with track and thumb

The track represents the range of possible values and the position of the thumb indicates the selected value. The only method of interacting with the slider is dragging the thumb along the track.

2.2.2 Prototypes

A prototype is a quickly developed version of the user interface that may be incomplete, inefficient, or simply a simulation of interacting with the system. Prototypes may be used for demonstration, evaluation or testing of the user interface without necessarily going to all of the implementation effort. Prototypes may be presented to users to develop an understanding of how an interface will look and behave (Nielsen, 1993; Snyder, 2003) and to collect feedback for refining the design (Rudd, Stern, & Isensee, 1996).

There are two general types of prototypes: low-fidelity prototypes which are constructed quickly with little or no functionality, and high-fidelity prototypes which are fully interactive but require more development time. Actual fidelity varies according to breadth of features, level of functionality and refinement of appearance (Virzi, Sokolov, & Karis, 1996).

Low-fidelity prototypes

Low-fidelity prototypes are quickly constructed versions of the system's user interface providing limited or no functionality, thereby costing little to produce. They can be used to generate ideas about how the user interface might look and behave (Rudd et al., 1996). The implementation may be as little as sketches on paper, a method known as 'paper prototyping'. Or it may be quickly mocked up screen layouts with little functionality actually implemented or little refinement of appearance. They can be used to demonstrate concepts, design alternatives, screen layouts and for collecting requirements. However, low-fidelity prototypes are not useful for details such as navigation or interaction (Rudd et al., 1996).

High-fidelity prototypes

High-fidelity prototypes are fully interactive versions of the system's user interface. They are quickly developed on the computer, although the implementation may simply be a simulation of the functionality (Rudd et al., 1996). They are not as quick and easy to produce, so are more costly, but they can provide much more complex information than low-fidelity prototypes. They can be used to investigate navigation, task flow, matching design and user models of a system, and the look and feel of the final product (Rudd et al., 1996). Ideally, the process should incorporate both low and high fidelity prototypes, with a progression from broad changes to fine tuning (Macleod, Bowden, Bevan, & Curson, 1997).

2.3 Navigation

For hundreds of years, humans have experienced the problem of "getting lost" in their geographic environment. Navigational aids have been developed to use spatial memory and other cognitive abilities of the navigator (Spence, 1999). These solutions include using maps, guides and landmarks to help the navigator build a "mental map" of the space (Benyon & Höök, 1997).

Navigation is "the cognitive process of acquiring knowledge about a space, strategies for moving through space, and changing one's meta-knowledge about a space" (Dahlbäck, 1998).

2.3.1 Information Spaces

Relatively recently, a new environment has developed for the exploration of electronic information spaces. This development is through advancements in computing technology, producing new ways to look at data. An information space is the concept of an environment in which exists all of the information that has been stored in a system (Benyon & Höök, 1997). This may exist in the physical world in a filing cabinet, or in the electronic world in a spreadsheet or database. This development has happened with increasing amounts of data becoming available in electronic forms and new technologies for working with those data. This produces a more virtual and abstract environment with its own challenges.

The differences between the physical and virtual spaces include that the virtual information space does not have a "stable Euclidean geometry" (Dahlbäck, 1998), navigation in information spaces is "relatively unconstrained" (Benyon & Höök, 1997), and virtual spaces do not contain "the explicit or implicit information that [movement is] in the right direction" (Dahlbäck, 1998). These characteristics add to the confusion of navigating an information space, and "getting lost" has been identified as a severe impediment in information spaces (Dillon, Richardson, & McKnight, 1990; Spence, 1999). Therefore, navigation in an information space has been recognised as one of the most important factors in user interface design (Benyon & Höök, 1997).

2.3.2 Visualising an Information Space

Humans take in most of their information visually. The brain is well developed for this with visual processing and reasoning occurring in the short term memory. However, the short term memory has limited capacity and short duration. Information quickly decays or gets replaced by other information (WanAdilah, NorLaila, & Rasimah, 2003). Visualisation creates external representations of data that are stable and can be translated by the brain to different scenarios (Crapo, Waisel, Wallace, & Willemain, 2000). These external representations can be used to assist the short term memory by reinforcing information.

For the purpose of this research, a visualisation of a two-dimensional information space will be considered. Typically, the visualisation is larger than the screen on which it is to be displayed so only a window of the visualisation is visible at one time (Guiard et al., 2004) (see Figure 6).

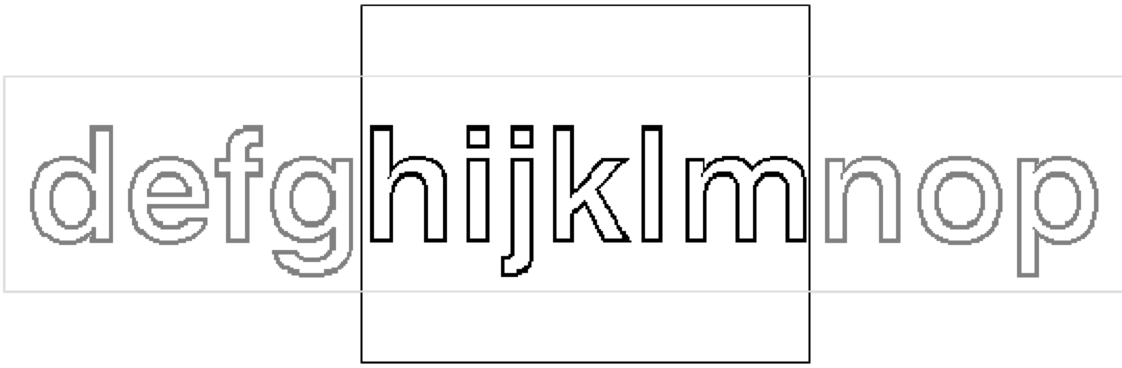


Figure 6 – An example visualisation, which is larger than the screen

To view other parts of the visualisation, the window may be moved within the visualisation to provide a different view, this action is known as panning (Spence, 2001).

Alternatively, the level of magnification may be changed to fit more or less of the visualisation into the window, known as zooming (Spence, 2001) (see Figure 7).

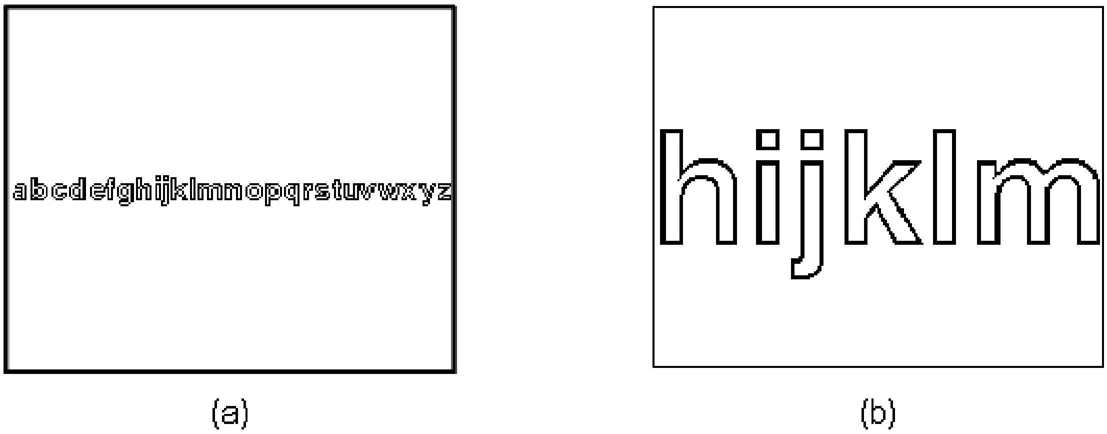


Figure 7 – An example visualisation at (a) low level of magnification and (b) high level of magnification

Successive views using zooming can reveal previously hidden detail (Jog & Shneiderman 1994). This can be further enhanced through use of a technique known as semantic zooming in which the way the data is displayed changes at different levels of magnification (Bederson & Hollan, 1994; cited in Card, Mackinlay, & Shneiderman, 1999), e.g. labelling elements with text that at a low level of magnification would not be visible or helpful (Loraine & Helt, 2002).

2.3.3 Navigational Strategies

The limited window through which an information space visualisation can be viewed causes problems because of the lack of contextual information (Leung & Apperley 1994), i.e. information about the size of the space and current position within that space. There are four main navigation strategies that developers may use to display large information spaces that address this to varying degrees. These navigation strategies will now be discussed.

Detail-Only

The Detail-only strategy uses a single view that can be panned over the detailed visualisation (see Figure 8). This strategy is beneficial only when the visualisation is not much larger than the view. It is not appropriate for comparing distant parts of a visualisation or when an overview is required (Plaisant, Carr, & Shneiderman, 1995).

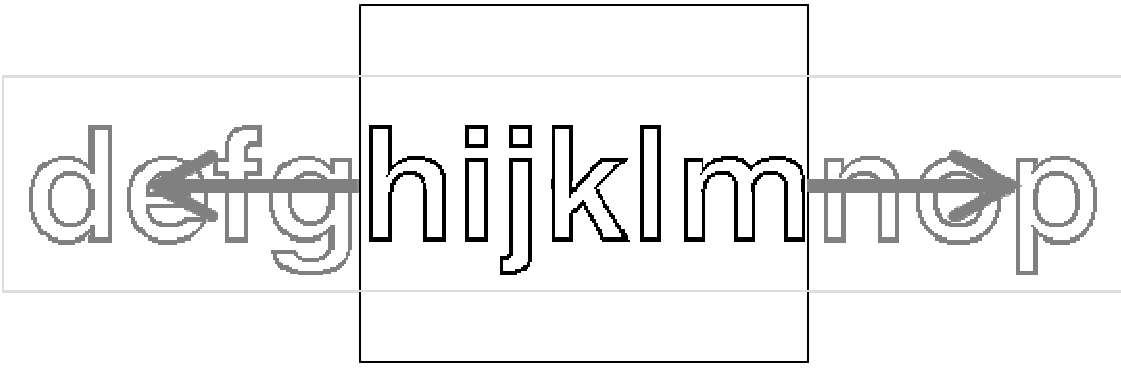


Figure 8 – Example of the Detail-Only navigational strategy

Zoom + Pan

The Zoom + Pan strategy uses a single view that has controls for both panning and zooming (see Figure 9). This enables the user to move between the levels of an overview to detail, with the magnified view replacing the original. It is an efficient use of screen space, however maintaining context can require frequent switching between detail and overview (Plaisant et al., 1995; WanAdilah et al., 2003).

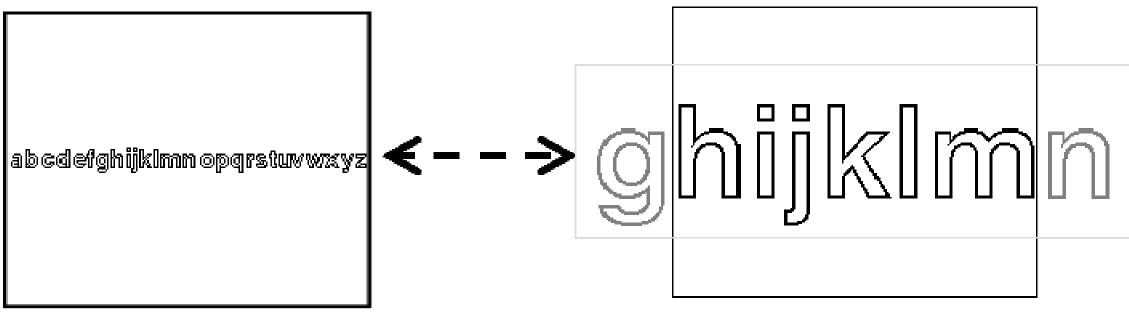


Figure 9 – Example of the Pan + Zoom navigational strategy

Overview + Detail

The Overview + Detail strategy uses two separate views, an overview and a detailed view (see Figure 10). Overview + Detail allows users to both examine details and see an overview, thus providing context (WanAdilah et al., 2003).

Figure 10 – Example of the Overview + Detail navigational strategy

The detailed view shows a window of the overview at a higher level of magnification. The window is marked on the overview to provide context. Showing the user the whole visualisation reduces the need to search through the detail and aids the user in choosing their detailed view (Plaisant, Milash, Rose, Widoff, & Shneiderman, 1996). The overview itself may give the user the information that they need, saving the user work. The size of the overview influences how much information can be displayed (Hornbæk, Bederson, & Plaisant, 2002) but reduces the space available for detail.

The two views can be connected, or "coupled", so that moving the window in the overview or panning the detailed view updates the other view (Plaisant et al., 1995). This coupling allows more efficient navigation as the user may navigate larger distances more quickly using the overview than if they used the detailed view.

Focus + Context

The Focus + Context strategy uses a single view with a distortion technique to display detail in an expanded region within the overview (Card et al., 1999) (see Figure 11). The expanded region may be interactively moved to view details as required while always maintaining the spatial relationships (Leung & Apperley, 1994). The main benefit of Focus + Context is that it provides local detail and global context at the same time, however this seems to rarely be a true requirement (Gutwin, 2005).

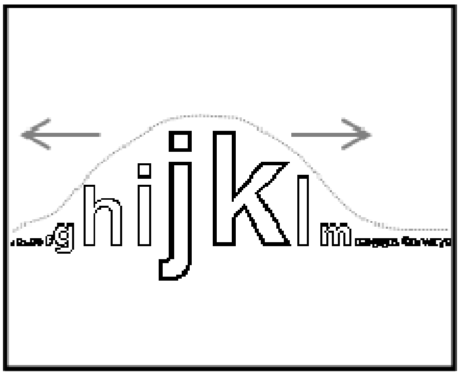


Figure 11 – Example of the Focus + Context navigational strategy

Examples of this strategy are the fisheye view (Furnas, 1986; cited in Card et al., 1999), and the Table Lens (Rao & Card, 1994; cited in Card et al., 1999).

2.3.4 Navigational Activities

Three generalised navigational activities that users perform in information space have been defined by Benyon and Höök (1997):

- **Exploration**
  Browsing a new space simply to see what objects exist in that space, and to develop an idea of how those objects are related (Spence 1999), not necessarily to meet any specific goal, e.g. browsing a book to see the range of chapters, tables and illustrations within it.
- **Wayfinding**
  Browsing to reach a specific location, also developing a mental model of the relative spatial locations of objects in the space (Spence 1999), e.g. browsing a book to find page thirty-two, a specific quote or an illustration.
- **Identifying Objects**
  Browsing to understand information about a generalised group of objects, particularly finding groups of specific types of objects and outliers from those groups, e.g. browsing a book to see how many illustrations were used or whether any illustrations use colour.

Movement through the information space may involve all of these activities, with the user switching between the activities to meet their needs (Doerry, Douglas, Kirkpatrick, & Westerfield, 1997a).

During these various activities users frequently become lost or disoriented (Otter & Johnson, 2000), and maintaining context while performing different activities has been described as one of the "most severe navigational difficulties experienced by users" (Doerry et al., 1997a).

2.3.5 Assisting Navigation

There are three main ways of supporting the user during navigation of a visualisation: use of metaphors, helping the user to build a mental model, and animation during navigational activities.

Use of Metaphors

Metaphors can be used by matching the appearance and behaviour in a new system to those of objects the user is familiar with. Users will then understand the new system to behave in a similar way (Karabeg, Akkøk, & Kristensen, 2004), e.g. the scrollbar (the metaphor for scrolling through a document) looks the same and can be expected to behave the same from application to application. By using common metaphors, users will be able to use existing knowledge to help better understand the information space (Dillon et al., 1990; Otter & Johnson, 2000). Developers need to understand how users think about the information being dealt with and be aware of existing metaphors (Nahum & Ward, 2001). This knowledge should be used to inform their designs.

Mental models

Helping users to understand and recognise their location in the information space will assist in navigating that space (Helt, Lewis, Loraine, & Rubin, 1998; Nielsen, 1990). A mental model represents a user's understanding of the information – the terminology, classifications, processes, and relationships of objects in that space (Lauesen, 2005; Tidwell, 2005) – and provides a frame of reference for navigation (Otter & Johnson, 2000).

The act of building a mental model of the information space helps the user understand the information represented (Spence, 1999). A mental model is created by using an information space; longer periods of use may help the user to capture deeper implications of the data into their mental model (Crapo et al., 2000). The more accurate the model, the less the likelihood of the user feeling lost (Otter & Johnson, 2000). Developers can support the users' development of mental models by using terminology, metaphors and representations of relationships that match what the user will understand.

Animation

Animation is a smooth and continuous transition between states. Appropriate animation can assist users to remain oriented (Spence, 1999), particularly when navigating spatial data (Bederson & Boltman, 1998; WanAdilah et al., 2003). A sudden change to the view can inadvertently confuse the mental model the user has developed and cause them to lose their sense of position and context (Spence, 1999). However, an animated transition allows the human eye to track objects without thinking about it (Bederson & Boltman, 1998). Shneiderman (1996) has found that, particularly in the action of zooming, animation helps users "preserve their sense of position and context". By taking these factors into consideration during the development process the user is less likely to become lost while navigating the visualisation.

2.4 Usability

Usability is a characteristic of a user interface that indicates the ease for users to understand and achieve their goals. This is more than "a warm, fuzzy feeling of 'user friendliness'" (Nielsen, 1993). The attributes of usability are focused on the user experience, and principles for usability and user-centred design activities have been developed to ensure that users' needs and specifications are understood and applied during development. The benefits of increased usability are manifold, including saving users time, more satisfied users and increased accessibility (Bevan, 1999). Users save time because they are quickly able to understand the system, the system behaves as they expect so less training is required, and they experience fewer errors (Doerry et al., 1997a; Lauesen, 2005).

Increased usability can provide a more satisfying user experience. If a user does not feel that a system is easy to use or relevant to their work, they are less likely to use it (Morris & Dillon, 1997) or are reluctant to use it, and they tend to blame themselves when they are unable to correctly use a system (Norman, 1988). Increased usability also makes the information that the system provides accessible to more people (Lauesen, 2005). Accessibility includes users' ability to understand how the system can be used to gain that information. If using the system confuses the user, the information is not accessible either (Doerry, Douglas, Kirkpatrick, & Westerfield, 1997b).

2.4.1 Attributes of Usability

There are different definitions of usability (Öörni, 2003), but four attributes generally associated with usability are (Bevan, 1999):

- **Learnability**
  The ease for novice users to understand the system.
- **Efficiency**
  The speed at which the user can complete their goals.
- **Effectiveness**
  The users' ability to complete their goals accurately and completely.
- **User satisfaction**
  A positive attitude towards the use of the product.

A usability problem is any issue that hinders a user's ability to achieve their goals. The usability problems within a system need to be identified during development to improve the usability (Lauesen, 2005). Usability needs to be considered in the context of the user, the task they are performing and their working environment (Abratt, Mallinson, & Bekker, 2003; Karat, 1997; Bevan & Macleod, 1994).

2.4.2 Principles for Usability

From a user's point of view the user interface is the system (Norman, 1988) and so the user interface has been the focus of a set of principles for usability. In 1994 Nielsen surveyed usability principles in the literature. At that time there were thousands of rules to follow which was seen as unmanageable (Nielsen & Molich, 1990). As a consequence, Nielsen produced a new base set of usability principles that aimed to explain the majority of usability problems found in "real systems".

Nielsen's (1994a) reduced set of usability principles are explained below:

1. **Visibility of system status**
   By looking, the user should be able to tell the state of the system (Norman, 1988). The user should be kept informed about what is happening with the system by being provided status information. Feedback should show that user input has been received by a cursor change or an animated response showing the action beginning (Snyder, 2003).

2. **Match between system and real world**
   The learning curve for the user to understand the system should be kept low. Allow the user to work with their innate knowledge by using terminology and metaphors that match the user's language and conceptual model. This principle also applies to the intrinsic properties of user interface components such as orientation of controls, for example the orientation of a scrollbar should correspond to the action that it causes (Norman, 1988).

3. **User control and freedom**
   The user should feel that they can safely explore the system and still maintain control (Tidwell, 2005). The user should easily be able to try an action, reverse that action and try something else (Tidwell, 2005). This includes ensuring that undo and redo actions are supported, and that controls such as sliders and scrollbars have the appropriate granularity or degree of movement (Snyder, 2003).

4. **Consistency and standards**
   The user should be able to transfer skills from one system to another (Nielsen, 1988). Utilising past experience and knowledge of the user minimises their memory load (Snyder, 2003; Otter & Johnson, 2000) and means that they intuitively know where to look for controls and how to use them (Tidwell, 2005).

5. **Error prevention**
   Ideally the user should not encounter errors. By recognising where errors may occur the system can be designed to prevent them.

6. **Recognition rather than recall**
   The user should be able to recognise all potential actions of the system. The user's mental model may decay due to lack of use (Spence, 1999) or the user may be interrupted during use (Snyder, 2003). When the user returns to the system their options should be apparent.

7. **Flexibility and efficiency of use**
   Regular users should have options in how they perform tasks and be able to use shortcuts for common tasks. The user should be able to work as they choose and not be unduly restricted in options for interaction (Snyder, 2003).

8. **Aesthetic and minimalist design**
   The user can be overwhelmed or distracted by too much information on the screen at

once (Snyder, 2003). A simple user interface is often better than a complex one, especially for new users or those who will not need all the functionality the system can provide. Instead, the most important information should be presented up front and the rest made available with a simple interaction (Tidwell, 2005).

9. **Help users recognise, diagnose and recover from errors**
   If the user does encounter an error, the system needs to help the user recognise that an error has occurred, diagnose what that error is and recover from it. Where errors cannot be prevented, the designer should prepare for that possibility.

2.4.3 User-Centred Design

Developers tend to have difficulty producing products with usability in mind (Doerry et al., 1997b). User-centred design addresses this by focussing on users' needs and specifications throughout the development process. The goal of user-centred design is an "easy to use and simple to learn system" (WanAdilah et al., 2003) emphasising that "the purpose of the system is to serve the user" (Soderston & Rauch, 1996).

For user-centred design, the developer must understand the users and their environment (Dahlberg & Götesson, 2004). Each user may differ in their age, gender, cultural background, computer literacy, work environment, roles, goals and needs (Kuniavsky, 2003). Each user is unique, so the compromise in the purpose of user-centred design is to find out what is "generally true" (Tidwell, 2005).

User-centred design includes activities and techniques which can be used to identify and document user needs and requirements (Bevan, 1999). These activities need to take place from the very beginning and throughout development to address questions about users' needs as they arise (Kuniavsky, 2003). User-centred design supports usability and development that meets user needs and specifications. Soderston and Rauch (1996) note that there has been an emphasis on functionality in the development of systems, but not on the users of the system. They explain that it had been assumed that users would be flexible enough to adapt to the system.

Through user-centred design, developers can understand what actually affects the user: their goals, tasks, terminology, mental models, skills and attitudes. Development can then be directed to those areas that will actually affect the user (Tidwell, 2005). Taking user needs into account will further support accessibility. Often the terminology and information structures used confuse the user (Bevan, 1999) and so the information is not "cognitively accessible" (Doerry et al., 1997b). Understanding the users' terminology will address this. User-centred design also supports a developer's understanding of the system (Dahlberg & Götesson, 2004) through the requirement for them to relate the system to users.

2.4.3.1 User-Centred Design Activities

The development process must include activities to investigate the needs and requirements of users. Bevan (1999) lists four user-centred design activities:

1. **Understand and specify the context of use**
   Specify the target users, their capabilities, the task being performed and the environment of use.
2. **Specify the user and organisational requirements**
   Specify the functionality of the system and requirements for interacting with the system.
3. **Produce design solutions**
   Produce potential design solutions by examining established designs and those of related disciplines, and the experience and knowledge of users.
4. **Evaluate designs against requirements**
   Present those designs to users, collect user feedback and use to improve the designs.

These activities should be repeated until the principles of usability are met (Bevan, 1999; Riihiaho, 2000; Whittaker, Terveen, & Nardi, 2000). Iterating in each step during development will ensure that questions about user needs and requirements are addressed as they arise (Kuniavsky, 2003).

2.5 Usability Evaluation

Usability evaluation is the assessment of the "ease with which users can learn and use a product" (Abratt et al., 2003) and is an essential part of human-centred design (Bevan, 1999; Riihiaho, 2000). Evaluating the system in this way ensures that the system meets the users' needs (Whittaker et al., 2000). The purpose of usability evaluation is to detect potential usability problems to be addressed before the product is released (Lauesen, 2005). The scope of usability evaluation includes the appearance of the interface, its functionality, its behaviour and other factors affecting usability (Macleod et al., 1997). The detection of usability problems before the system is released allows them to be addressed (Lauesen, 2005; Nielsen & Landauer, 1993), resulting in a more usable system. Re-work can be reduced as problems can be eliminated earlier without having written any code (Abratt et al., 2003). Usability evaluation shows whether users understand the user interface as the designers expect them to (Kuniavsky, 2003), refines understanding of how actual users perform their tasks, and detects when users are likely to have difficulties. Addressing these problems before the product is released increases user satisfaction and efficiency.

Due to the sensitivity of usability evaluation methods at identifying design weaknesses, some of the problems raised in the usability evaluation process may not pose a problem in actual use (Nielsen & Molich, 1990). However, by rating the severity of problems found, work can be prioritised and serious problems addressed. Overwhelmingly, usability experts report that even small evaluation trials have helped to identify potential usability problems, even before the final system is produced (Riihiaho, 2000; Snyder, 2003).

2.5.1 Methods of Usability Evaluation

There are many different methods in practice for the evaluation of usability. Ivory and Hearst (2001) list thirty-nine different methods, which they categorise as:

- **Testing methods,** where representative users are observed performing tasks with the system.
- **Inspection methods,** where an expert evaluates the system.
- **Inquiry methods,** where the focus is understanding the user environment and user needs.
- **Analytical Modelling methods,** where predictive analysis is used; and
- **Simulation methods,** where user interaction is mimicked to understand the time and number of interactions required to perform a task.

The detail of several usability evaluation methods relevant to this research are described later in the section.

It is common practice that a combination of usability evaluation methods be used. Each usability evaluation method serves a different purpose, with different strengths giving different insights into the usability of the system (Kuniavsky, 2003) and revealing different problems (Kelley & Allender, 1995; Riihiaho, 2000). Also, as usability has a diverse collection of qualities (see Section 2.4.1), to capture more of these qualities usability must be examined in many ways (Kuniavsky, 2003; Riihiaho, 2000; Karat, 1997; Kelley & Allender, 1995).

While participant involvement in usability evaluation is desirable, it is not always practical (Abratt et al., 2003). Participant involvement can be expensive and difficult to coordinate (Riihiaho, 2000) so more informal and less structured methods (such as usability testing and heuristic evaluation) are more frequently used than formal and structured methods (such as focus groups and cognitive walkthrough) (Vredenburg, Mao, Smith, & Carey, 2002).

A common recommendation is the combination of a usability inspection method, usually heuristic evaluation, with a usability testing method to provide the most thorough coverage (Riihiaho, 2000; Kelley & Allender, 1995).

A usability evaluation can be performed with a single evaluator, however several participants are recommended (Nielsen & Landauer, 1993). Typically, a single evaluation will identify only a proportion of the usability problems. By aggregating the results from multiple evaluations, the proportion of problems found is increased (Nielsen & Molich, 1990). Statistically, five evaluations will give a level of accuracy sufficient for many studies (Nielsen, 1993).

Usability evaluation should take place throughout the development process, starting before prototypes are prepared and continuing through to the development of the final product (Riihiaho, 2000; Macleod et al., 1997). The level of change that can occur will be dependent upon the stage of development. The more effort that developers have invested in developing a system, the less willing they are to change it (Lauesen, 2005). In earlier iterations, where little implementation has occurred, usability evaluation can lead to broad changes. Later in the development process, where significant implementation has occurred, usability evaluation should only be used for fine tuning the design. By including usability evaluation in each iteration, a progression from broad changes to fine tuning can be used (Macleod et al., 1997).

The three methods of usability evaluation which form part of this research (Usability Inquiry, Usability Inspection, and Usability Testing) are now examined.

2.5.2 Usability Inquiry

Usability inquiry focuses on understanding the user environment and user needs. These methods should be used at the start of the development process, for selecting the target users for the product and investigating what those users actually do and what is valuable to them (Kuniavsky, 2003).

A common usability inquiry method is contextual inquiry, where representative users are interviewed in their work environment (Riihiaho, 2000). The information collected includes ethnographic data, the tools that they use, the tasks they perform, the sequences in which actions are performed and the ways that information is used (Kuniavsky, 2003; Riihiaho, 2000).

Federoff (2002), who was examining game developers, found that the developers felt uncomfortable with an evaluator looking over their shoulder all day. Also, as developers spent much of their time working alone at a computer, the developers felt little was to be gained by watching their typing. It was found that having the developers think about their activities out loud was most comfortable and productive, however only two of the five developers did so.

Other usability inquiry methods are: using questionnaires and surveys to collect information about users, focus groups for structured discussion between multiple users, observing users in their work environment, and examining logged data of a users' interaction with a system (Ivory & Hearst, 2001).

Usability Inquiry methods will help understand the mental models of users, the tools, terminology and methods that they use, and their goals and values (Kuniavsky, 2003). Understanding this, usability principles can be better met, users' mental models can be utilised, and consistency and existing standards can be applied in development.

2.5.3 Usability Inspection

Usability inspection is where an expert evaluates the system. It can be performed quickly and very early in the development process, even before any prototypes are prepared (Riihiaho, 2000).

The most common usability inspection method is heuristic evaluation, where an expert evaluates a user interface design for conformance to a set of usability principles (Riihiaho, 2000; Nielsen & Landauer, 1993; Nielsen & Molich, 1990). The evaluator uses the usability principles as a guide to direct their focus. One set of principles is described in

Section 2.4.2, 'Usability Principles'. The typical heuristic evaluation will take a single evaluator one-to-two hours, during which the evaluator will make several iterations through the individual elements in the design (Nielsen, 1994). Where the design does not conform to the principles, this is identified as a potential problem (Riihiaho, 2000; Nielsen, 1994).

Usability inspection methods have the benefit that they can be used early in the development process (Nielsen & Molich, 1990) so that most principles can be assessed without the involvement of users or an actual user interface (Riihiaho, 2000). The method produces few false positives (Nielsen & Molich, 1990). However, usability inspection methods are not guaranteed to find every single usability problem in a system (Nielsen, 1994), nor do they generate solutions to those problems found (Cockton & Woolrych 2002). Also, the probable quality of any redesign cannot be predicted (Nielsen, 1994). Nevertheless, heuristic evaluation is frequently used (Riihiaho, 2000) and is found to be an efficient and effective method of usability evaluation (Bevan & Macleod, 1994).

2.5.4 Usability Testing

Usability testing involves observing representative users performing tasks with the system. They help develop a deep understanding of users' abilities and expectations of the system and are good at discovering serious problems that users may encounter (Kuniavsky, 2003).

Usability trials are structured interviews where a participant is observed performing given tasks with the system (Kuniavsky, 2003; Riihiaho, 2000; Nielsen & Landauer, 1993). In usability trials, the participant may be interacting with a prototype of the system or a section of the system, rather than the completed product. To be meaningful, a usability trial must include participants who are representative of the target users, include tasks that are representative of the system and be performed in conditions representative of the normal use conditions (Kuniavsky, 2003; Bevan, 1999). Snyder (2003) reports that most usability trials in general are either one or two hours in length.

Usability testing provides insight into the mental model and working methods of real users (Ivory & Hearst, 2001; Nielsen & Landauer, 1993), immediately showing whether users understand the user interface as the designers expect them to (Kuniavsky, 2003). It also provides information about usability that cannot be collected any other way (Bevan & Macleod, 1994).

However, usability testing is very expensive as the method cannot be performed before any development has occurred and it is a time-consuming process (Jeffries, Miller, Wharton, & Uyeda, 1991), both in performing the trials and the analysis. Also, usability trials are not statistically representative. “If three out of four people say something, that doesn’t mean 75% of the population feels that way. It does mean that a number of people may feel that way, but it doesn’t mean anything numerically” (Kuniavsky, 2003).

2.5.4.1 People Involved in a Usability Trial

A usability trial should usually involve several people: the participant who will attempt to perform the tasks, a facilitator who conducts the trial session and one or more observers who make notes and record data during the session. One person may take several roles in the trial (Riihiaho, 2000).

The participant needs to be a representative user (Bevan & Macleod, 1994). They are the sole interactor with the system during the usability study.

The facilitator is responsible for the smooth running of the usability trial (Riihiaho, 2000), guiding the participant through the study and encouraging feedback from the participant. However, the facilitator needs to remain neutral towards the system and the designers, and not influence the feedback of the participant (Snyder, 2003).

Observers do not participate in the running of the usability trial, instead observing the participant as they perform their tasks and recording the participant’s intellectual and emotional responses (Webb et al., 2005). As the observer’s role is subjective in nature, it is useful to have more than one observer to reduce bias (Ivory and Hearst, 2001).

2.5.4.2 Protocol of Usability Testing

Kuniavsky (2003) states that “there are two goals in conducting user interviews: getting the most natural responses from evaluators and getting the most complete responses”. The second goal can be achieved by the facilitator prompting the participant to speak by asking questions; however, to maintain natural responses, the facilitator needs to be careful how they prompt the participant (Kuniavsky, 2003).

A common technique for performing usability testing is the “think-aloud protocol” (Snyder, 2003; John & Marks 1997). As the participant performs the given tasks, they are encouraged to speak about their thoughts and experiences. This enables the researcher to understand the user's mental model and the problems they are experiencing (Virzi et al., 1996).

To ensure the most complete responses are expressed, the facilitator should ask the participant about nonverbal responses and about their experience (Kuniavsky, 2003). The facilitator should also ask questions to understand the mental model of the user and the problems that they are experiencing (Virzi et al., 1996). It can be difficult to get natural responses though, as people have been conditioned to place the blame on themselves when they have difficulty understanding the system or make errors (Kuniavsky, 2003). This will limit the amount and usefulness of feedback provided. To set the participant at ease, and thereby get the most natural responses, the participant needs to be assured that errors are viewed as the fault of the system, not the participant (Nielsen, 1993), that they can freely criticise the system (Kuniavsky, 2003), and that anonymity is preserved (Nielsen, 1993).

During the trial, the participant must be allowed to work through the tasks unaided, to reflect the normal conditions of use. This needs to be explained at the start of the evaluation, and any questions addressed then (Snyder, 2003; Riihiaho, 2000; Bevan, 1999).

Finally, task selection and specification is important for the validity of the study. The tasks that the participant performs during the trial must be representative of those they would perform in their actual working environment (Nielsen, 1993). If not, the results will not necessarily be valid to their actual working environment (Riihiaho, 2000; Whittaker et al., 2000).

2.5.4.3 Data Collection

Data collection is a crucial part of any research (Webb et al., 2005). There are many ways of collecting data during the usability study, including: video capture of the computer screen, video or audio capture of the participant during the trial, notes taken by the observer(s), and automated capture such as keystroke information (Ivory & Hearst, 2001).

Video capture allows the participant to work undisturbed and can be used for detailed analysis after the trial (Bevan, 1999).

Data can be captured automatically from user interface events where the participant interacts with the user interface, such as whenever a button is clicked, a text field is entered, or a control, such as a scrollbar, is used (Hilbert and Redmiles, 2000; Rudd, Stern et al., 1996).

Notes taken by the observers are often the best source of information from a usability study, as they give human feedback about the intellectual and emotional experiences of the participant during the study (Öörni, 2003), such as noting where the user seemed confused, frustrated or satisfied.

The data collected during the study can be collated and analysed to describe the participant's experience: their successes, misunderstandings, mistakes and opinions (Kuniavsky, 2003). The results from multiple individual trials should be aggregated (Nielsen & Landauer, 1993), collecting the most common observations into lists of functionality and presentation problems (Kuniavsky, 2003). The analysis should lead to recommendations for improving the usability of the system (Riihiaho, 2000).

2.5.5 Measures of Usability

In order to evaluate usability, a way to measure usability within a usability trial is required (Öörni, 2003; Bevan & Macleod, 1994). These measures can be used to compare alternative designs, identify where problems occur and where improvements can be made (Kuniavsky, 2003; Bevan & Macleod, 1994).

The attributes of usability were defined in Section 2.4.1. If these are to be useful, qualitative measures for these are needed. It is recommended that multiple types of objective and subjective measures be collected because agreement and consistency among results will enhance the credibility of the findings (Riihiaho, 2000; Whittaker et al., 2000).

There are two broad categories of measurable data that can be collected during usability evaluation: objective measures, quantitative measurements of the use of the system; and subjective measures, qualitative assessments of how much the participant likes the system (Nielsen & Levy, 1994).

2.5.5.1 Objective Measures

Objective measures are quantitative measurements of each participant's use of the system (Kuniavsky, 2003; Nielsen & Levy, 1994). Some common objective measures include the time taken to perform a task, the number of errors made performing that task and how many participants complete the task successfully (Kuniavsky, 2003; Ivory & Hearst, 2001; Nielsen & Levy, 1994). Due to the small numbers of participants typically in a usability study, timing events is not recommended as the statistical error will be greater than the accuracy of the data a stopwatch could record (Kuniavsky, 2003).

Instead of absolute measurements, it is recommended that a relative measurement range for each metric be created (Kuniavsky, 2003; Öörni, 2003; Macleod et al., 1997). Kuniavsky (2003) cites the practice of a usability evaluation consultancy, which uses the following range to measure how long people take to perform a task:

0 – Fail

1 – Succeed very slowly in a roundabout way

2 – Succeed a little slowly

3 – Succeed quickly

This level of accuracy will be sufficient for comparative analysis and the average results will provide a way to compare tasks to each other and between designs (Kuniavsky, 2003; Öörni, 2003). Such measures can be created for each of the usability factors being evaluated. As these measures are open to subjectivity, the criteria need to be decided upon before starting assessment (Macleod et al., 1997) to ensure objectivity and consistency.

2.5.5.2 Subjective Measures

Subjective measures are qualitative assessments of how much each participant likes the system (Nielsen & Levy, 1994), using their opinions and preferences to determine the perceived usability and acceptability of the system by its potential users (Bevan & Macleod, 1994). This is an important consideration, as objective measurements and user satisfaction are not necessarily correlated (Lauesen, 2005; Whittaker et al., 2000; Macleod et al., 1997). A system may be efficient to use but not very satisfying to use, or vice versa.

Common ways of conducting a subjective evaluation include having participants rate the system on one or more rating scales, select the design they liked best, or rank their preferences of all demonstrated designs (Nielsen & Levy, 1994). Likert Scales (Jamieson, 2004) are frequently used to indicate the degree of agreement with statements about the designs (Nielsen & Levy, 1994). The most common range of the rating scales is 1-5, with higher numbers signifying more positive attitudes.

By prompting for participant satisfaction under several categories, the participant is encouraged to give a broader perspective on their experience (Roth, 1999), and more detailed and useful feedback. "Instead of simply saying 'This software is not good', the [participant] can say *how* the software is no good" (Roth, 1999).

2.5.5.3 Severity of Usability Problems

The problems found during the evaluation can be classified by their scope and severity (Riihiaho, 2000). There are different ways by which this can be done. The severity of a usability problem can be measured by three factors: the frequency with which the problem occurs, the impact of the problem if it occurs, and the persistence of the problem (Nielsen, 1997).

Different scales are used by different usability experts. One example, used by Nielsen (Nielsen, 1997), presents a five point rating scale for usability problems:

0 – I don't agree that this is a usability problem at all

1 – Cosmetic problem only: need not be fixed unless extra time is available

2 – Minor usability problem: fixing this should be given low priority

3 – Major usability problem: important to fix, so should be given high priority

4 – Usability catastrophe: imperative to fix this before product can be released.

The problems should then be presented in order of severity (Nielsen, 1992).

2.6 Application to Bioinformatics

In this section the principles of the previous sections will be examined in the context of bioinformatics.

The main objective for users of bioinformatics software is the handling and presentation of genetic sequences and their annotations (Molidor, Sturn, Maurer, & Trajanoski, 2003). As biologists work with a genetic sequence, they develop new questions about the data and need to explore the data and investigate it further. The challenge for bioinformatics is to make this data "usable and accessible for biological discovery" (Molidor et al., 2003).

2.6.1 User Interface

As well as humans being very visually orientated, biology itself is an "intensely visual science" with information stored not as mathematical equations or as algorithms, but summarised visually as diagrams and illustrations (Helt et al., 1998).

In 2003 an exploratory study investigated the display of genetic sequences on screen for the purpose of exploration and discovery (Rutherford, 2003). It became evident at that time that the length of the data and maintaining context during navigation were problems.

One visual representation for displaying genetic sequences and their annotations, guiding developers on how to assist biologists, has been described by Lorraine and Helt (2002). The basic visualisation consists of the sequence being displayed along a single linear axis. As the data is one-dimensional, the axis perpendicular to this is used to display information related to that sequence such as annotations (Loraine & Helt, 2002) (see Figure 12).

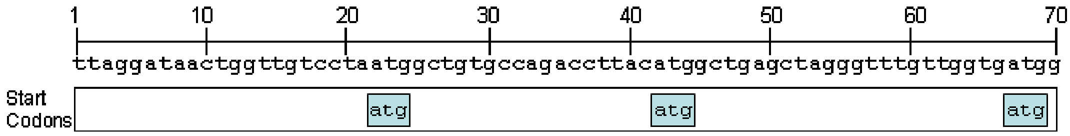


Figure 12 – A simplified example of Lorraine and Helt's visualisation scheme

2.6.2 Navigation

As stated in Section 2.3.2, humans take in most of their information visually, so providing a visual representation of DNA sequence data will assist in taking in and understanding that information.

Biologists need to interact with the visualisation in a more complex manner than a static image permits (Loraine & Helt, 2002). They will engage with all three of the navigational activities described in Section 2.3.4: exploration, wayfinding and identifying objects. In performing these they often need to modify the view: panning the visualisation and changing the level of magnification of the view.

Loraine and Helt (2002) also made recommendations for approaches to interactivity. They particularly encouraged the use of animated transitions and semantic zooming to show additional information when it is possible to do so, supporting the discussion in Section 2.3.5.

Three different approaches to magnification are:

Detail-view

The detail-view shows the nucleotides of the actual sequence, mapped to absolute positions on the sequence. By showing one linear axis, annotations can be mapped to their positions alongside the sequence (see Figure 12).

At this level of magnification only a window of the sequence can be seen, other parts of the sequence may be seen by panning the view, as discussed in Section 2.3.2.

Overview

The overview is a visual representation of the entire sequence. Helt et al. (1998) use a representation based on the pattern of light and dark bands when a DNA sample is prepared and observed under the microscope, with which biologists will be familiar. Another representation is to display large features of the sequence as shaded boxes (Loraine & Helt, 2002) (see Figure 13).

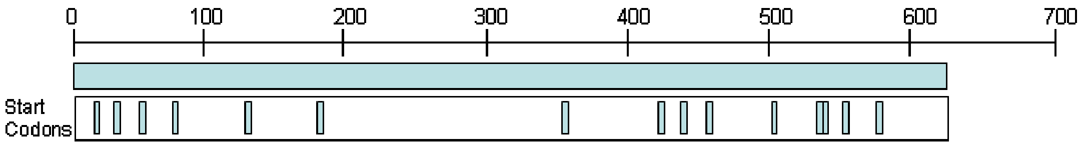


Figure 13 – A sample sequence overview showing a ruler along the horizontal axis, a representation of the DNA sequence, and boxes marking the locations of possible Start Codons

At locations relative to their actual position in the sequence, graphical objects such as boxes can be used to represent them (Helt et al., 1998).

At this level of magnification the whole sequence can be seen within the view however detail cannot be seen. Detail may be accessed by zooming-in, as discussed in Section 2.3.2.

Multilevel Views

The user is able to change the magnification level of the view to intermediate levels between the detail-level and overview-level, with features being shown at the appropriate level. At these intermediate levels, both panning and zooming are options for changing the view. Further discussion on these strategies was in Section 2.3.3.

Loraine and Helt (2002) found that although most genomic browsers provided some mechanism for one-dimensional zooming, none provided interactive or dynamic zooming. These techniques would assist the user during navigation (Section 2.3.5).

2.6.3 Usability

Bioinformatics application development has largely been by biologists-turned-programmers or vice versa and usability has often not been well understood or emphasised (Fuchs, 2000). Studies have confirmed that the learning curve for new users is steep (Javahery, Seffah & Radhakrishnan, 2004) and that users can be overwhelmed by the difficulty of regular tasks such as annotation (Liu et al., 2003). Thus, the investigation of usability and accessibility of bioinformatics applications is essential (Roos 2001).

There is growing recognition for the need to examine usability of applications in bioinformatics as is noted by Fox et al. (2007) in their review of tools, databases and resources for life sciences research, and growing interest in the use of usability principles in the development of bioinformatics applications. For example, Chen et al. (2007) indicate that they intend to study user experiences for the purpose of improving usability, and Hearst et al. (2007) also indicate plans to perform usability testing. However there remains to be some education on what usability is and how it can be measured, for example Forment et al. (2008) describe their application EST2Uni as "user-friendly" and "user-oriented", however the subjects of their user research were themselves.

An appropriate approach to identifying areas for improvement of usability of bioinformatics applications could be based on the user-centred design principles discussed in Section 2.4.3 and the methods for evaluating usability discussed in Section 2.5.

2.7 Summary

This chapter began with an introduction to genetic sequences and DNA, the source of data that bioinformatics researchers work with. The following sections discussed factors that apply to this data.

The section on user interfaces discussed approaches for interactivity and common user interface components and options for prototyping. Next, strategies for viewing large information spaces, navigation activities within them and ways of assisting navigation were discussed. Then, the fundamentals of usability were introduced, getting beyond "user-friendliness" to the attributes and principles of usability, and how usability may be evaluated and user needs and expectations incorporated into the software development process. Each of these factors is well understood in its own right but there appear to be gaps in the application of these bodies of knowledge to bioinformatics. Clearly there is a need for research into the usability of bioinformatics software.

Chapter 3: Proposed Research

As discussed in Section 2.6, biologists need to explore their data, including genetic sequence data, in a more rich and interactive manner. However, there has been little consideration of them as users or of usability in the development of bioinformatics applications. Many bioinformatics applications claim to be “user-friendly” (e.g. Hu, Frith, Niu & Weng, 2003; Vernikos, Gkogkas, Promponas, & Hamodrakas, 2003; Forment et al., 2008) however this is rarely tested (MacMullen & Denn, 2005). As a result, users may find the learning curve too steep and it has been recommended that the investigation of usability and accessibility of bioinformatics applications is essential (MacMullen & Denn, 2005; Roos, 2001).

This research will investigate the needs of bioinformatics users, the extent to which those needs are met by current applications, and how those needs could be better met.

3.1 Target Users

The target user of this research is any researcher working with genetic sequences on a computer. More specifically, the research is for researchers in a small organisation who are scientists and lab technicians, such as those at Crop and Food Research Ltd, Lincoln. It is expected that these researchers will vary in their domain knowledge and computer experience, how involved they are in day-to-day activities, and the tasks that they are performing.

3.2 Research Aim and Objectives

This research aims to develop and evaluate ways of displaying and navigating a genetic sequence that are usable by the users described above.

This aim will be met by using the user-centred design activities described in Section 2.4.3.1. The research will consist of the following stages which will be covered in subsequent chapters of this thesis:

1. **Define the target users and their goals**
   A questionnaire will be used to investigate the target users, their work environment and the tasks that they perform (Chapter Four).

2. **Examine current bioinformatics applications for existing designs**
   Current bioinformatics applications will be examined to identify existing designs for displaying the information and facilitating navigation (Chapter Five).
3. **Perform a usability evaluation of identified design elements**
   Heuristic evaluation will be used to critically analyse the designs identified in the preceding chapter. From this, a smaller set of designs will be chosen for further evaluation (Chapter Six).
4. **Develop prototype designs and perform a usability study using them**
   Prototypes of the selected designs will be developed (Chapter Seven) and be presented to users in usability testing (Chapter Eight).
5. **Analyse results of usability testing to produce recommendations**
   Data collected during the usability study will be analysed and the results and discussion presented (Chapter Nine). Overall conclusions from the research will be drawn and future investigations suggested (Chapter Ten).

3.3 Scope

This research will be limited to dealing with data for single DNA sequences in the scale of thousands of bases, as this length is typical of sequences being used in biological research (Luscombe et al., 2001). Single sequences will be considered only, as working with multiple sequences adds a significant level of complexity.

The research will be restricted to the facilities for navigation of the sequence. Benyon and Höök (1997) see the issue of navigation through the information space as one of the most important in interactive system design. Also, of the seven tasks that Shneiderman (1996) lists for information spaces, it is contended that gaining an overview and then zooming to detail always come first. Thus, these interactions must be correctly implemented to facilitate the later tasks.

Two usability evaluation methods will be used to assess the designs: heuristic evaluation, and usability testing. This combination is as recommended in Section 2.5.1 for providing the most thorough assessment. High-fidelity prototypes (discussed in Section 2.2.2) will be developed for usability testing because the focus of this research is on navigation, so a fully interactive prototype will be required for assessing usability with users.

3.4 Summary

The outcome of the research will be recommendations for the designs incorporated into bioinformatics applications for the purpose of navigating genetic sequences and their annotations, based on users' needs and requirements. It is also hoped that this research will encourage further use of usability evaluation in the development of bioinformatics applications.

Chapter 4: Defining the Target Users

To design with the users in mind, it must be known who the users are and what they do. A questionnaire was used to understand who they are and the environment in which they work. The questionnaire covered the methods and the tools that participants use to work with genetic sequences, and the tasks that they typically perform.

Ideally, development of a computer application would begin with a contextual inquiry, as described in Section 2.5.2 User Inquiry Methods. However, as Federoff found (2002), such observation and interviewing would be more intrusive and intense than required. Instead, a questionnaire was used to define the context of use and to support user-centred design.

This chapter describes the content of the questionnaire and how it was distributed. The results are also presented and the information gained discussed.

4.1 Content of the Questionnaire

The questionnaire was designed to collect similar information as a contextual inquiry whilst taking a relatively small amount of each participant's time. The questionnaire is shown in Appendix A.

The questions were in three main categories: Demographics, Tools, and Tasks.

The questionnaire was distributed both on paper and electronically to people at the Germplasm Enhancement Division of Crop and Food Research Ltd. Distribution was widespread, with recipients encouraged to forward it to others. Respondents were self-selected and as so are not necessarily representative of all users.

4.1.1 Demographics

This section was intended to find out the background of participants. This information also provided a basis for understanding each participant's later responses. Questions relate to:

- the participant's position in the organisation (i.e. student, technician, scientist, or senior scientist),
- how often the participant uses a computer, and
- the participant's experience with computers, with bioinformatics data, and with using a computer for carrying out bioinformatics tasks.

4.1.2 Tools

The purpose of this section was to understand the way that the participants currently work with genetic sequences. Questions relate to:

- how the participant uses computers to store the genetic sequences and results from bioinformatics applications that they have obtained in research,
- how often the participant uses specific bioinformatics applications and how well the participant feels that they are able to use those applications,
- how frequently the participant chooses to work with bioinformatics data on paper and why, and
- examples of bioinformatics applications that the participant has found easy to use and examples of programs that the participant has found frustrating to use.

4.1.3 Tasks

The purpose of this section was to understand the bioinformatics tasks that participants typically perform. This was to direct the focus of this research and give insight into the ways that genetic sequence data is used.

Participants were asked to describe "Two tasks that they perform with genetic sequences" and given a template for their responses.

Each heading of the template had an explanatory statement and a sample response. The headings and accompanying statements were:

- Goal, the task you wish to perform,
- Inputs, the information you enter into the tools,
- Outputs, the resulting information you are after,
- Method, the sequence of tools that you use, and
- Problems, is there something about this task that you find frustrating or unnecessarily time consuming (if any)? Please explain.

4.2 Findings from the Questionnaire

Eleven people responded to the questionnaire. As it was primarily distributed electronically, the response rate is not known. This was a small response, but the range of responses contained valuable information. The results are discussed below.

4.2.1 Demographics

Question 1. Job description

The expected response was a job position, such as student, technician, scientist or senior scientist. While some participants gave a job position, others gave a brief description of what tasks they performed (e.g. "Marker design", "Vector construction" or "Transgenic plant production and analysis"), or the crop that they work on (e.g. "barley" or "potato"). The additional information was interesting, but not of practical use.

While a job position was not always stated, the categories shown in Table 1 were deduced from the information given.

Table 1 – Responses to Question 1, 'Job Description'

Job Description	**Tally**
PhD Student	2
Senior scientist / Research leader	6
Scientist	2
Technician	1

This seems to be heavily weighted in senior positions however there was ambiguity in the assignment of categories and provided the participant is still active in research and they provide valid and varied responses, this should not impact the project.

Question 2. Computer use

A summary of the responses to this question is shown in Table 2.

Table 2 – Responses to Question 2, 'Computer Use'

Computer Use	**Tally**
Less than once a week	0
Several times a week	0
Every day	0
Several times a day, or most of the day	11

All participants said that they use a computer daily, so lack of familiarity with computers is unlikely to be an issue.

Question 3. Computer experience

A summary of the responses to this question is shown in Table 3.

Table 3 – Responses to Question 3, 'Computer Experience'

Computer Experience	Tally
I can use the World Wide Web (WWW) to search for information	11
I can use a word processor to produce formatted documents	11
I can use a spreadsheet to perform simple calculations	11
I can produce reports with a database	6
I can design tables and queries for a database	3
I can perform simple programming tasks	1
I can perform advanced programming tasks	0

All participants are capable of performing basic tasks and using common applications. Experience performing higher-level tasks, such as designing database tables, was less common.

Question 4. Work with genetic sequence data

A summary of the responses to this question is shown in Table 4.

Table 4 – Responses to Question 4, 'Work with genetic sequence data'

Work with genetic sequence data	Tally
Less than once a week	4
Several times a week	5
Every day	2
Several times a day, or most of the day	0

Experience with working with genetic sequence data varied. There was not enough data to draw definite conclusions, but these results do not seem to be biased by the high number of senior scientists who responded.

Question 5. Computer use for bioinformatics tasks

A summary of the responses to this question is shown in Table 5.

Table 5 – Responses to Question 5, 'Computer use for bioinformatics tasks'

Computer use for bioinformatics tasks	**Tally**
Less than once a week	5
Several times a week	4
Every day	2
Several times a day or most of the day	0

Experience with using a computer for bioinformatics tasks was varied. In this case, senior scientists used computers for bioinformatics tasks the least often. Seven respondents reported the same usage as for Question 4, suggesting that when they do work with genetic sequences it is on a computer. An irregularity was that three respondents reported less usage of a computer than for bioinformatics data, as they had reported in Question 4.

4.2.2 Tools

Question 6. Work on paper for bioinformatics tasks

A summary of the responses to this question is shown in Table 6.

Table 6 – Responses to Question 6, 'Work on paper for bioinformatics tasks'

Work on paper for bioinformatics tasks	**Tally**
Always	0
Most often	2
Sometimes	8
Never	1

Most participants acknowledged working on paper at least some of the time. The two who responded "Most often" were senior scientists, and the participant who responded "Never" was a student who worked with genetic sequence data less than once a week.

Question 7. Reasons to work with data on paper

The most common reason, given by five participants, was to be able to add notes and markings to (i.e. annotate) the sequence. Four participants stated that they keep a hard copy of the computer output for their own records. Other reasons were that the participant didn't know how to perform the tasks on computer, to compare outputs from different sources, because of difficulty reading on-screen, and to be able to count bases (to determine locations).

Question 8. Storing genetic sequences

Participants were asked to indicate all methods that they used to store genetic sequences obtained in their research, and to circle their preferred method. A summary of the responses to this question is shown in Table 7.

Table 7 – Responses to Question 8, 'Storing genetic sequences'

Storing genetic sequences	Tally Used	Tally Preferred
in a text file	5	3
other: DNAMAN	5	3
in a database	4	1
in a spreadsheet	4	1
in a Word document	4	1
on paper (printed)	4	0
in Genbank	2	1
other: Sequencher	1	1

A wide range of methods were reported for storing genetic sequences, and there was no consistent pattern of use or preference. DNAMAN and Sequencher are additional applications reported by participants that were not in the original list. Most respondents reported using several types of storage.

Question 9. Storing results from bioinformatics applications

Participants were asked to indicate all methods that they used to store results from bioinformatics applications, and to circle their preferred method. A summary of the responses to this question is shown in Table 8. A wide range of methods was reported for storing results from bioinformatics programs. There was some preference for Word, probably as it retains formatting and graphics during the copy-and-paste operation. Notably, no one preferred to use paper, but almost everyone did use paper. Again, most respondents reported using several types of storage.

Table 8 – Responses to Question 9, 'Storing results from bioinformatics applications'

Storing results from bioinformatics applications	Tally Used	Tally Preferred
on paper (printed)	8	0
in a Word document	6	4
in a text file	5	1
in a spreadsheet	4	2
other: DNAMAN	2	2
in a database	2	1
other: Mega	1	1

Question 10. Use of bioinformatics applications

Participants were asked to indicate how often they used tools from a list of bioinformatics applications. A summary of the responses to this question is shown in Table 9.

Table 9 – Responses to Question 10, 'Use of bioinformatics applications'

Application	Often	Sometimes	Total
Genbank	7	3	10
BLAST	5	6	11
Primer3	4	4	8
EMBOSS		1	1
ORF Finder		2	2
Repeat Masker			0
VecScreen		2	2
Other: DNAMAN	3	2	5
Other: NEB Cutter		1	
Other: Sequencher	1	1	2
Other: SeqScape	1		
Other: Spidey	1		
Other: ABI Sequence Analysis	1		

Almost all participants reported some experience using the bioinformatics applications Genbank and BLAST. Other applications named as being frequently used were DNAMAN, Sequencher, ABI SeqScape, ABI Sequence Analysis, NEB Cutter and Spidey. These applications are included in the examination of applications in Chapter Five.

Question 11. Ability with bioinformatics applications

Participants were asked to indicate how well they thought they could use bioinformatics applications. A summary of the responses to this question is shown in Table 10.

Table 10 – Responses to Question 11, 'Ability with bioinformatics applications'

Application	Very well	Enough to be useful	Poorly
Genbank	3	5	2
BLAST	2	7	2
Primer3	2	6	
EMBOSS			1
ORF Finder	1	1	
Repeat Masker			0
VecScreen	1	1	
Sequencher	1	4	1

Most participants reported understanding how to use Genbank, BLAST, and Primer3, otherwise there did not appear to be great confidence in the use of any application. A small number of participants reported poor understanding of how to use Genbank, BLAST, EMBOSS and Sequencher.

Question 12. Examples of useful software

The responses for this question were not as expected, possibly because the question was not clear enough. The expected response was an explanation of a useful feature of an application, however participants often gave an application name only, or the example wasn't software related.

Applications named as being easy to use were SeqScape, Sequencher, Primer3, and BLAST. Only one application, SeqScape, was recommended by a participant specifically for ease of use. Again, these applications are included in the examination of applications in Chapter Five.

Question 13. Examples of frustrating software

The responses for this question were not as expected, for the same reason as Question 12. Applications named as being difficult to use were Primer3, DNAMAN and MEGA. These were most often for difficulties annotating the sequence and file management. One participant described their “hassle” with DNAMAN, an application they reported using “Often”. Another participant said “I struggle with everything”, also reporting that they worked with their data on paper “most often” and stored their data either on paper or in text files.

4.2.3 Tasks

Some tasks described by participants that were within the scope of the research include:

- displaying a sequence with annotations for marker design,
- displaying a sequence with annotations to facilitate mapping, and
- displaying a sequence with annotations to clone a sequence into a plasmid.

A common sub-goal of all these tasks (identified by the researcher) was the need to bring all information into one view.

Some tasks described by participants were not sufficient for analysis due to lack of detail, problems were outside the scope of this research or it was not clear how performing the task could be made easier because the researcher lacked understanding of the process. The methods that participants described were generally by using the applications discussed in earlier sections. In three instances participants specifically mention using print-outs.

Many problems identified by participants were related to biological factors which are outside the scope of this research. However relevant problems identified were that:

- there is no single software package for their tasks,
- there is a lack of control over output from applications,
- it is difficult to annotate sequences,
- sharing data between applications is difficult,
- they don’t know how to use the tools.

By applying usability methodology, bioinformatics applications developers would be able to avoid many of these problems.

4.3 Conclusions

A questionnaire was used to better understand the target users and their working environment. Although only a small number of users responded, the responses contained enough valuable information to assist with understanding the users' environment. The participants were found to all have experience using a computer, and reported being capable of performing basic tasks and using common applications. It was confirmed that users sometimes choose to work with bioinformatics data on paper due to difficulties working with the data on computer. There were no predominant methods for storing genetic sequences or data from bioinformatics applications. A list of commonly-used applications was collected from participants, and descriptions of commonly-performed tasks and problems encountered.

A common need seems to be viewing a sequence and its annotations, showing the proposed research to be relevant to these users. Some of the problems encountered were that the applications did not meet all of their needs, they did not know how to use the application, and difficulty applying annotations to a sequence.

From these findings, the user environment is better understood. The next chapter examines a sample of existing bioinformatics applications, including those named by participants of this questionnaire, to discover what designs already exist for navigation of genetic sequences.

Chapter 5: Examination of Current Applications

Existing bioinformatics applications were examined to see how they facilitate navigation of genetic sequence information. The applications are those that questionnaire respondents reported using (see Chapter 4) as well as other bioinformatics applications in the public domain. A list of the designs for navigating genetic sequences is produced to look for any predominant approach and for further investigation. Several applications that perform similar tasks will also be examined to see if there are effective solutions outside of the bioinformatics domain.

5.1 Applications Examined

There are many bioinformatics applications and sequence viewers available. The applications examined included those named by questionnaire respondents: the Genbank database (accessed through NCBI Map Viewer), DNAMAN, BLAST, Primer3 and Sequencher. Other applications were found by a literature search for bioinformatics applications. More applications could also have been examined but were felt to add nothing more.

Where possible, applications were installed and examined, otherwise the examination was dependent upon research reports and documentation. If there was not sufficient information to understand the designs used, the application was not included in this examination. Each application was examined for the navigation strategy used (discussed in Section 2.3.3) and the designs utilised for providing navigation through panning and zooming. A summary of the applications examined and navigational designs is provided in Appendix B.

5.1.1 APIC (Bisson and Garreau, 1995)

APIC is an Application Programming Interface (API) for browsing and navigating genomic data, and was developed as a visualisation interface to be used in other projects. It can operate as a standalone application but has also been customised for use in applications such as Imagene (Medigue, Rechenmann, Danchin, & Viari, 1999).

One multilevel view is provided which can be panned and zoomed. Additional views may be added to show either the same or another sequence and these views may be interconnected or not. Views that are connected affect each other so that interactions with one view are reflected in the other views. The transitions for panning are smooth however transitions for zooming appear not to be.

Panning is provided by scroll bar. Buttons are also provided to move to next/previous feature or by a specified distance, both sets labelled ‘p’ for previous and ‘n’ for next. Zooming is provided by buttons labelled ‘Zoom in’ and ‘Zoom out’ that zoom by a fixed ratio.

5.1.2 Apollo (Lewis et al., 2002)

Apollo is a desktop application that was developed to provide an interactive tool for viewing and evaluating sequence data and annotations. It was initially developed for annotating fruit-fly DNA sequences but is claimed to be extensible to any genomic data.

One multilevel view is provided which can be panned and zoomed from overview-level to detail-level. For panning, scroll bars are provided and buttons to move by a fixed amount that are labelled ‘<’ and ‘>’. For zooming, buttons are provided that zoom in or out by increments (‘x10’, ‘x2’, ‘x.5’, or ‘x.1’). The transitions for panning are smooth however transitions for zooming are not.

5.1.3 Artemis (Rutherford et al., 2000)

Artemis is a Java application for interactive sequence visualisation and annotation. The developers state that Artemis is suitable for sequences from the size of small genes to whole genomes.

Two multilevel views are provided; both show the sequence and each can be individually zoomed from overview-level to detail-level. The initial configuration is an Overview in the upper view and Detailed View in the lower view. Views are not interconnected. A list of features and details is displayed textually in a third view.

Panning of the individual views is provided through the use of a scroll bar. Zooming is provided by a vertical scroll bar. Artemis was the only application examined that utilised smooth transitions for zooming, however only with a limited number of levels of magnification so transition between these levels is not completely smooth.

5.1.4 BLAST (McGinnis & Madden, 2004)

BLAST is an application with a web interface for comparing a sequence to a database of sequences to find similar sequences. BLAST is one of the most heavily used bioinformatics applications in the public domain (McGinnis and Madden, 2004).

Two static views are provided in the results of a BLAST search, a graphical Overview and textual detail of the alignments found. The Overview shows a representation of the submitted sequence as a reference and parallel to that are possible alignments. The Overview is connected to the Detailed View so that clicking on a result in the Overview will hyperlink to the detail. The detail is a multiline textual display.

The textual detail, which is displayed in multiple lines across the screen, can be panned by scroll bar. No other level of magnification is available. This is not an interactive experience though, but a static web page.

5.1.5 ChARMView (Myers, Chen, & Troyanskaya, 2005)

ChARMView is a Java application for viewing data mapped to chromosome locations to display gene expression data against the location of the gene on the genetic sequence.

A multilevel view of the chromosome and related information is provided, initially displaying an Overview. This may be zoomed to intermediate levels, but is not intended for the purpose of viewing the detail-level of the sequence.

Panning is provided by scroll bar. Zooming is provided by Marquee Tool or Magnifying Glass Tool (see Section 2.2.1.1, Direct Manipulation Approaches). The zooming mode is by selecting a mouse zoom mode from a selection of buttons (Figure 14) and then zoomed by direct interaction with the view. However, the view is updated without use of smooth transitions.

Figure 14 – Mouse tools for selecting level of detail in ChARMView. From left to right: Marquee tool, Zoom-in tool, Zoom-out tool, Fit-in-window

5.1.6 DNAMAN (Woffelman, 2004)

DNAMAN is desktop sequence analysis software package with a number of functions for performing biological investigations with genetic sequences.

Several different displays are provided in DNAMAN. The only display relevant to this research was the “Sequence Map” display which contains a multilevel view, initially displaying an overview, and a detailed view that can be panned only.

Panning is provided in both views by the use of scroll bars. The Overview may be zoomed by clicking on buttons in the toolbar that zoom in or out by a fixed amount. The transition for panning is smooth however zooming is stepwise and not smooth.

5.1.7 Ensembl (Hubbard et al., 2002)

Ensembl is a web-based application for working with complete genomic sequences and their features. Annotations are automatically generated for a wide range of analyses which become relevant at different levels of magnification.

Four static views are provided, each at a different level of magnification. There is an overview, detailed view, and two views in-between. Each view is interconnected so that the selection of a location in one view is reflected in all other views. Each view may be panned and zoomed however these transitions require an entire page reload in which all views are refreshed. The interface is very ‘space-intensive’ – it takes several screens to view all the information provided and it is not possible to display a global overview and detail on the screen at the same time.

Panning is provided through buttons to pan a fixed distance in a direction. For zooming, there are buttons to zoom in/out, labelled ‘+’ and ‘–’, and a control to select a level of zoom (see Figure 15).

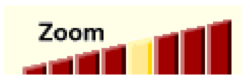


Figure 15 – Control for selecting level of magnification in Ensembl. Currently the middle level of magnification is selected.

As any change requires a complete refresh of the web page, smooth transitions are not supported.

5.1.8 GeneViTo (Vernikos et al., 2003)

GeneViTo is a Java application for exploring genomic data with annotations from diverse sources, particularly for working with genomic functional elements and their products.

The interface consists of a view in which an aggregated representation of the sequence is displayed in multiple lines across the screen, a small Circular Map for selecting a region in the sequence (Figure 16), and panels containing detailed information about the features in the sequence. 280,000 base segments of the sequence are displayed at a time.

Figure 16 – Circular map for selecting location in the sequence in GeneViTo

Panning is provided through use of the Circular Map control. Clicking on a location in the Circular map control or a feature in the Overview changes the Detail view to show that location. There is no facility for changing the level of magnification and the transition for panning is not smooth.

5.1.9 GAP (Bonfield, Smith, & Staden, 1995)

Genome Assembly Program (GAP) is a Unix application with a Tcl/Tk interface. It is an early bioinformatics application focussing on providing an interactive graphical user interface. It was designed for working with multiple DNA sequences at once and the developers claim that it is useful for large and small projects.

Many different display configurations are provided but only the Template Display and the Join Editor were relevant to this research. The Template Display is a multilevel view that may be panned and zoomed. The Join Editor is a detail-level view that may be panned only. These displays cannot be viewed simultaneously.

In the Template Display, panning is provided by a scroll bar. Zooming is provided by Marquee Tool technique (see Section 2.2.1.1, Direct Manipulation), which is used by selecting a mode ("zoom in" or "zoom out") and dragging the mouse over the view to select a region which is then zoomed-in or zoomed-out. The Join Editor also provides panning by scroll bar and has buttons for panning as well, labelled '<<', '<', '>' and '>>'. No other levels of magnification are available in this view. The transitions for these displays are not smooth.

5.1.10 Genotator Browser (Harris, 1997)

Genotator is a desktop application for performing batch analysis of genetic sequences to build a database of annotations. Genotator Browser has a Tk interface to view the sequence and those annotations.

Two displays are provided, a Map Display that displays a multilevel view that can be panned and zoomed; and a Sequence Display that provides detail-level view that may be panned only. These displays cannot be viewed simultaneously.

In the Map Display, panning is provided by scroll bar and zooming is provided by a horizontal slider. In the Sequence Display, panning is by scroll bar and no other levels of magnification are available. How transitions are presented in Genotator was not evident.

5.1.11 Gestalt (Glusman & Lancet, 2000)

Gestalt is a web-based tool for predicting the location of genes in a sequence, and is claimed to be useful for sequences from thousands to millions of bases long. Gestalt produces a static image that is an overview of the entire sequence and its annotations.

One static view is produced which is an overview of the entire sequence and tracks displaying information about the sequence such as statistical prediction quality and features. The user may click on a feature in the image for the web page to reload with some detail about the selected feature. No facility for panning or changing the level of magnification is provided.

5.1.12 MEGA (Kumar, Nei, Dudley, & Tamura, 2008)

Molecular Evolutionary Genetics Analysis (MEGA) is a desktop application for a broad range of tasks working with genetic sequences including editing sequence data, performing alignments between sequences and conducting comparative analyses. From the beginning, MEGA was designed for working with evolutionary data and provides statistical methods for these. It has evolved over the years from a character-based interface to a graphical user interface.

The Sequence Data Explorer provides one detail-view of the sequence. Tracks containing other aligned sequences are displayed beneath the main sequence. Panning is provided by scroll bar and occurs smoothly however there is no other level of magnification.

5.1.13 NCBI Map Viewer (Wheeler et al., 2005)

NCBI Map Viewer, a web-based application for viewing entire genomes, is the main interface for viewing the data of the National Center for Biotechnology Information (NCBI) institute GenBank sequence database. This application is intended for searching for specific features which are then shown in this interface, rather than browsing and exploration.

Two views are provided; an overview of the sequence and a detailed list of features. NCBI Map Viewer shows the sequence vertically, which is unlike most viewers which show the sequence running horizontally. Detailed information about features is available via hyperlinks, which display in separate web pages.

The overview operates as a Connected View, providing both panning and zooming. Clicking on a location on the overview presents a context menu from which the user may choose to "Recenter" on that location (i.e. pan), "Zoom in" (by various magnitudes) or "Zoom out". There is also a control to select a level of zoom (see Figure 17).

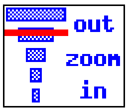


Figure 17 – Control for selecting level of magnification in NCBI Map Viewer

Any change requires a complete refresh of the web page, so smooth transitions are not presented.

5.1.14 NEBcutter (Vincze, Posfai, & Roberts, 2003)

NEBcutter is a web application for viewing the locations on a sequence where a restriction enzyme (a biological tool) would recognise and operate on the sequence.

NEBcutter has two distinct levels of magnification. First, an Overview of the sequence is displayed on which a location can be selected to be viewed in detail. The Detailed View may be panned or can return to the Overview.

In the Overview, panning is not available. A location may be seen in detail by marking the location on map and clicking a button labelled 'Zoom'. This new view is loaded in the Detailed View.

In the Detailed View, the sequence may be panned by buttons (confusingly under the heading 'Zoom'), labelled '<<' and '>>'. Clicking on the button labelled 'Unzoom' returns to the Overview. These views are displayed separately and no smooth transitions are used.

5.1.15 Primer3 WWW Interface (Rozen & Skaletsky, 2000)

Primer3 is a tool for designing primers, a type of feature that is used in making copies of a sequence. A web interface has been developed for Primer3 results. Exploration and detailed examination of results is important as biologists come to Primer3 with a variety of problems and need to find a solution to match their problem.

The Primer3 output is displayed as a 'quasi-graphical' overview showing the sequence, the location of the primer features and the region of the sequence that will be copied; and list of detail about each of the sets of features. The overview is 'quasi-graphical' because it is textually composed however the location of textual elements is meaningful.

This is a static web page which can be panned by scroll bar, but no other level of magnification is available.

5.1.16 RegulonDB (Salgado et al., 2001)

RegulonDB is a Java application for viewing the circular *Escherichia coli K-12* genome, a simple organism with a relatively small genome. The data presented in RegulonDB is circular, rather than linear, reflecting the reality of the sequence.

RegulonDB offers two distinct displays. The first display is a Circular Map of the genome which cannot be panned or zoomed, the second display is a Detailed View. Clicking a location on the Circular Map opens a new window showing the Detailed View for the selected location.

Panning of the Detailed View is provided by buttons labelled with arrows, '<--' and '-->'. No other level of magnification is provided. These views are displayed separately and no smooth transitions are used.

5.1.17 SeqScape (Applied Biosystems, 2004)

SeqScape is desktop software for advanced sequence analysis that is designed for integration with sequencing software also produced by Applied Biosystems.

The "Assembly Display" shows an Overview of the sequence above a Detailed View. The views are interconnected so clicking a location in the Overview displays that region in the Detailed View and panning the Detailed View reflects the new region selected in the Overview.

The Detailed View may be panned by scroll bar. No other level of magnification is provided. Panning by scroll bar provides a smooth transition, however by Connected View there is no smooth transition.

5.1.18 Sequencher (Gene Codes Corporation, 2003)

Sequencher is a desktop application for working with sequences from sequencing machines to produce longer contiguous sequences.

Two relevant views are provided, an Overview and Base View. These are not available simultaneously. In the Overview, panning is not available. In the Base View, a textual display of the sequence is provided in multiple lines across the screen. Changing between the views is provided by a button that toggles from 'Overview' to 'Base View'.

The Base View may be panned by scroll bar. As the sequence is in multiple lines across the screen, the scroll bar is vertical. The view is toggled from Overview to Base View by clicking a button. The panning of the Base View occurs smoothly, however there is no smooth transition for changing the level of magnification.

5.1.19 SeqVista (Hu et al., 2003)

SeqVISTA is a Java application for viewing sequence annotations integrated from diverse sequence analysis sources. SeqVISTA was developed to address some of the problems of bioinformatics applications identified by the application's developers: integrating data, navigating features and identifying exact locations on the sequence.

A Connected View is used, consisting of two views of the sequence: an Overview of the entire sequence and a Detailed View. A panel with information about features is also provided. These views are connected so that selecting a feature in the Overview or feature panel shows the associated section of the DNA sequence in multiple lines across the screen.

The Detailed View may be panned by scroll bar or by clicking on the section of the overview to be shown in the detail. There is no method of selecting another level of magnification. Panning by scroll bar provides a smooth transition, however by Connected View there is no smooth transition.

5.1.20 UCSC Browser (Karolchik et al., 2002)

The University of California Santa Cruz (UCSC) Genome Browser is a web site application for viewing and querying genomic sequences, initially developed for viewing the data of the Human Genome.

A Connected View consisting of two views of the sequence is provided: an Overview of the entire sequence and a multi-level Detailed View. The Detailed View can be zoomed from overview-level to detail-level.

The Detailed view may be panned by buttons or by clicking on a location in the overview. The panning buttons are labelled ‘<<<’, ‘<<’, ‘<’, ‘>’, ‘>>’ and ‘>>>’, each panning the sequence by a different amount. Other ways for panning are also available, but those above are the most immediately available.

Zooming of the Detailed View is also by buttons, to zoom in by a relative amount or to base-level. The zooming buttons are in groups for zooming in or zooming out, and are labelled ‘1.5x’, ‘3x’, ‘10x’. The user may also specify co-ordinates of the sequence to display by using a textbox labelled ‘position’ and then clicking a button labelled ‘jump’.

Altogether, twenty controls are provided for zooming and panning. This accommodates many different methods that a user could navigate a sequence by. As the interface is provided via a web page, any interaction requires a reload of the web page and as such smooth transitions are not supported.

5.2 Summary of Techniques

A range of techniques for navigation were found, a summary of the information collected from the applications examined is contained in Appendix B. The ways of providing panning and zooming were grouped and tallied. The results are show in Table 11 and Table 12.

Table 11 – Tally of instances of methods for providing panning in the applications examined

Design	**Typical Configuration**	**Tally**
Buttons	Two or four buttons, horizontal	8
Scroll Bar	Horizontal	14
Connected View	Overview displayed above detail	4
Circular Map	Overview displayed in corner of display	2
No interaction	N/A	1

Table 12 – Tally of instances of methods for providing zooming in the applications examined

Design	Typical Configuration	Tally
Buttons	Two buttons, horizontal, to zoom in or zoom out	2
	Two or more buttons, horizontal, zoom by amounts	3
	One button, toggling between overview and detail	1
Slider	One instance horizontal, one instance vertical	2
Select Level Tool	Horizontal, ordered detail to overview	2
Magnifying Glass Tool	Button to select mode, no visible control	1
Marquee Tool	Button to select mode, no visible control	2
No interaction	N/A	8

From these tables it can be seen that there is no predominant method of providing either panning or selecting the level of magnification, although buttons and scroll bars are commonly used.

Web-based applications all required the refreshing of the web page to update the views, meaning that the sequence could be navigated interactively but not in real-time or smoothly. This is the same in desktop applications if there is no transition between the old and new image. Only one of the applications examined above, Artemis, utilised animation for providing smooth transitions. However, even in Artemis the smoothness of transitions was limited by the number of discrete levels between levels of magnification, resulting in jumpy animation.

5.3 Other Applications

Other applications for exploring one- and two-dimensional data that provide panning and zooming were also examined to see if they offered any alternative solutions. The applications examined were: Adobe Acrobat (Adobe, 2004), ECan Online GIS (ECan, 2005), and Google Maps (Google, 2005). For each of these applications, the relevance of the application to this research will be justified and a description of the method for panning and selecting level of detail given. Relevant designs will be added to the list for further examination.

5.3.1 Adobe Acrobat Reader (Adobe, 2004)

Adobe Acrobat Reader is a desktop application for viewing PDF documents. The document is displayed as a one-dimensional sequence of pages, so the spatial arrangement is similar to the data type being considered in this research.

The main view of Adobe Acrobat Reader is a sequence of pages from a document. These pages may be zoomed from an overview to the detail of the page content. The sequence is vertical, unlike the typically horizontal arrangement of the sequence in bioinformatics applications. There are multiple methods of panning and zooming, presumably to account for the many different ways that users may wish to work. Panning is provided by scrollbar; clicking and dragging the view; or buttons that move to the previous page, next page, start or end of the document. Zooming is provided by selecting and using a mouse zoom mode: Dynamic Zoom, Marquee Tool or Magnifying Glass Tool; by buttons for zooming in, zooming out or zooming to a fixed level ("fit page width", "fit page height"); or by using a drop-down box to select a level of magnification (expressed in percentages).

5.3.2 ECan Online GIS (ECan, 2005)

ECan Online GIS is a web-based application for viewing Geographic Information Systems (GIS) data for the region of Canterbury, New Zealand. This is an implementation of the ArcIMS system (ESRI, 2005). Although GIS works with two-dimensional data there are similarities in navigation.

Panning is performed by selecting the mouse mode for panning, then clicking and dragging the view. Zooming is by selecting a mouse mode for zooming (see Figure 18): Marquee Tool or Magnifying Glass Tool, and then clicking on the location to zoom on.

Figure 18 – Mouse tools for selecting level of detail in ECan GIS Viewer. The first two buttons are zoom-in tool and zoom-out tool; the last button is panning tool

5.3.3 Google Maps (Google, 2005)

Google Maps is a web-based application that provides access to maps of world locations, with levels from an overview of the entire world map down to (in some regions) street level. As with the previous application, Google Maps works with two-dimensional maps and so there are similarities in navigation.

The view is a two-dimensional map. Panning is performed by clicking and dragging the view, or by using a set of arrow buttons. Zooming is performed using a slider that is positioned over the view (see Figure 19).

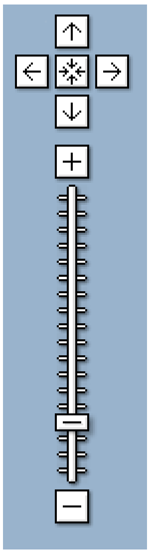


Figure 19 – Navigation controls for Google Maps. The controls at the top pan the view. The slider, '+' button and '–' button control level of detail

As a result of examining these other applications some alternative methods of navigation will be considered also: Hand Tool for panning by direct manipulation, Dynamic Zoom for zooming by direct manipulation, and On-View Slider.

5.4 Conclusion

Twenty bioinformatics applications were examined for methods of providing panning and selecting level of detail. Three other related applications were also examined. No predominant approach was found for either panning or selecting level of magnification, however a list of potential designs for each task was developed. These designs will be critically examined for conformance to principles of usability to determine suitability for user testing in the next chapter.

Chapter 6: Heuristic Evaluation

The designs identified in bioinformatics software applications in Chapter Five are now critically evaluated with heuristic evaluation, a usability inspection method, to form an objective opinion of each design. The result of the inspection is a list of usability issues with comments on any principles that each design violates. These issues will be ranked by severity to evaluate the suitability of each design. This evaluation will then be used to refine the list of designs to be subjected to further testing.

6.1 Evaluation Process

Heuristic evaluation is a critical evaluation of an interface for conformance to usability principles, described in Section 2.5.3. The evaluation was based upon Nielsen's heuristic principles (1994) which are listed in Section 2.4.2. These principles are well-cited and therefore considered a standard.

6.1.1 Context

The designs are considered in the context of how they could be used to interact with a sequence of which only a window can be viewed (a 'view', Section 2.3.2). This view has a range of positions within the sequence, from the very start to very end of the sequence. The view can be moved to see different sections of the sequence. The purpose of the panning design is to manipulate the position of the view within the sequence.

This view of the sequence also has different levels of magnification where more or less information will fit within the view. Levels of magnification range from the detail-level to overview-level. The purpose of the zooming design is to manipulate the level of magnification of the view.

6.1.2 Method

Each of the designs was evaluated individually by each of the heuristic principles. If a principle was not met adequately, the principle and an explanation were listed. For brevity, the principles that were met adequately are not listed.

The severity of each issue found was categorised in one of the following ways:

- A problem indicates that a principle will affect users.
- A warning indicates that a principle may affect users.

The severity of issues was categorised because some issues can be minimised through the implementation of the design, whilst other issues cannot be resolved.

Usually multiple evaluators would evaluate each design. However, as this evaluation is of elements of an interface, as opposed to an entire interface, it was felt that one evaluator was sufficient.

6.2 Evaluation of Designs for Panning

The evaluation of designs for panning is discussed below.

Buttons

In the bioinformatics applications examined there were either a set of two buttons (one for panning left and one for panning right) or four buttons (for panning left or panning right by different distances).

Type	Principle	Explanation
Problems	1. Visibility	No indication of system status (e.g. context or position) is given by the design.
	3. User control	User has little control over degree of action.
Warnings	2. Internal model	Layout and orientation of design elements will be important. Panning has a directional action so the orientation is logically horizontal.
	6. Recognition	Labelling will be important for users to recognise the purpose of each button.
	8. Minimalist design	Sufficient information may not be provided as minimum and maximum values are not displayed.

Scroll Bar

Scroll bars are a common user interface element for scrolling through a document that is larger than can fit in the screen (see Section 2.2.1.2 Scroll bars). In the bioinformatics applications, scroll bars were always horizontal, matching the action they performed.

Type	Principle	Explanation
Problems	3. User control	The number of values that can be selected on a scroll bar is limited to the discrete number of positions on it.
Warnings	2. Internal model	Layout and orientation of design elements will be important. Panning has a directional action so the orientation is logically horizontal.

Connected View

Two or more views of the same visualisation are provided, one view showing an overview, the other views showing detail (see Section 2.3.3 Overview + Detail). Selecting a region in the overview pans the detailed view to show that region.

Type	Principle	Explanation
Problems	3. User control	The number of values that can be selected on the overview is limited to the number of positions on it.
Warnings	2. Internal model	Layout and orientation of design elements will be important. Panning has a directional action so the orientation is logically horizontal.
	4. Standards	The design is not standard however similarity to scroll bars should make it predictable.
	6. Recognition	Recognition of how to use the design will be impacted by how the overview and interaction are implemented and displayed.
	8. Minimalist design	May provide more information than is required.

Circular Map

This is a variation on the Connected View where the Overview is distorted to be circular rather than linear.

Type	Principle	Explanation
Problems	2. Internal model	A Circular Map does not match the reality of a linear sequence, especially when ends of the sequence are brought together in the map and look much closer than they actually are.
	3. User control	Circular movement is more difficult than one-directional movement, so is likely more difficult to control.
	3. User control	The number of values that can be selected on the map is limited to the number of positions on it.
	4. Standards	The control is not standard and is not similar enough to standard controls to make it predictable.
Warnings	1. Visibility	The distortion in producing the Circular Map may obscure the data.
	6. Recognition	Recognition of how to use the design will be impacted by how the overview and interaction are implemented.
	8. Minimalist design	May provide more information than is required.

Direct Manipulation

This design allows the user to click directly on the visualisation and drag it to pan the view. There is no visible control (see Section 2.2.1.1).

Type	Principle	Explanation
Problems	1. Visibility	The only visible element is the cursor which does not provide indications of system status, context, or cue on how to interact with the interface.
	4. Standards	Panning by dragging appears in a number of applications now, however it may not be understood by all users without explanation. It was not found in any of the bioinformatics applications examined.
	6. Recognition	There is no cue on how to interact with the interface, so it will rely on recall.
	8. Minimalist design	There is no visible control, so no information is provided other than that in the view. This will not provide a sense of position to help maintain context.
Warnings	2. Internal model	The mapping of movements should match the direction the user wishes to move.
	3. User control	The interaction will need to be carefully designed so that the user can achieve what they want in a controlled manner (e.g. if the user wants to move a small distance rather than a large distance).

6.3 Evaluation of Designs for Zooming

The evaluation of designs for zooming is discussed below.

Buttons

In the bioinformatics applications examined there were either two buttons for zooming in and zooming out, or four buttons for zooming in, zooming out, going direct to overview-level and direct to detail-level.

Type	Principle	Explanation
Problems	1. Visibility	No information to help provide context is given by the design.
	2. Internal model	Zooming does not produce a directional response in the same way that panning does so an orientation cannot be logically assigned.
Warnings	3. User control	User has little control over degree of action.
	6. Recognition	Labelling will be important for users to recognise the purpose of each button.
	8. Minimalist design	This design may not provide sufficient information. This will not provide a sense of position to help maintain context.

Slider

Sliders are a common user interface element for selecting a value from a continuous range (see Section 2.2.1.2 Sliders). In the bioinformatics applications examined, one zoom slider was horizontal and in another the slider was vertical.

Type	Principle	Explanation
Problems	2. Internal model	Zooming does not have a directional response in the same way that panning does so an orientation cannot be logically assigned.
Warnings	3. User control	The number of values that can be selected on a slider is limited to the discrete number of positions on it.
	6. Recognition	Labelling will be important so that users can easily see the orientation of the mapping.

Select Level Control

This is a design from which a level of magnification is selected from a discrete set of levels. This may be considered as a special case of buttons (see Section 2.2.1.2 Buttons). For an example, see Figure 15 on page 50.

Type	Principle	Explanation
Problems	2. Internal model	There is no consistent way for the scale to be interpreted: some may see the small bars as representing showing smaller items (more detail) whilst others may interpret it as representing less detail (i.e. an overview).
	3. User control	There is only a limited set of levels available.
	6. Recognition	It is difficult to simply label buttons to indicate “detail” and “overview”. This means that the design relies firstly on recall and then on more complex labelling.
Warnings	4. Standards	The design is not standard; however familiarity with buttons will make interaction with the design predictable.
	6. Recognition	Labelling will be important for users to recognise the purpose of each button.

Dynamic Zoom

Dynamic Zoom involves clicking on the view and dragging up to zoom in, or dragging down to zoom out (see Section 2.2.1.1 Direct Manipulation). There is no visible control.

Type	Principle	Explanation
Problems	1. Visibility	The only visible element is the cursor, which does not provide indications of system status, context, or how to interact with the interface.
	2. Internal model	Zooming does not have a directional response in the same way that panning does so an orientation cannot be logically assigned.
	4. Standards	This design is currently found only in one mainstream application, Adobe Acrobat Reader, so cannot be considered 'standard'.
	6. Recognition	There is no cue on how to interact with the interface so the design will rely on recall.
	8. Minimalist design	There is no visible control, so no information is provided other than that in the view. This will not provide a sense of position to help maintain context.
Warnings	3. User control	The interaction will need to be carefully designed so that the user can achieve what they want in a controlled manner (e.g. if the user wants to zoom in a small amount rather than a large amount).

Magnifying Glass Tool

The Magnifying Glass Tool allows the user to click on a point in the view to zoom in on that point (see Section 2.2.1.1 Direct Manipulation). There is no visible control.

Type	Principle	Explanation
Problems	1. Visibility	The only visible element is the cursor, which does not provide indications of system status, context, or cue on how to interact with the interface.
	2. Internal model	While zooming-in using the Magnifying Glass Tool follows the metaphor of a magnifying glass in the real world, the metaphor does not follow for zooming-out but is used in many applications, e.g. Microsoft Office.
	6. Recognition	Whilst zooming in using the magnifying glass tool follows the metaphor of a magnifying glass in the real world, the metaphor does not follow for zooming out. Instead, the user must select a different mode to zoom out which, being separated from the tool (the mouse cursor), will require recall.
	8. Minimalist design	There is no visible control, so no information is provided other than that in the view. This will not provide a sense of position to help maintain context.
Warnings	3. User control	The interaction will need to be carefully designed so that the user can achieve what they want in a controlled manner (e.g. if the user wants to zoom in a small amount rather than a large amount).

Marquee Tool

The Marquee tool allows the user to click directly on the view and select a region of the view to zoom on (see Section 2.2.1.1 Direct Manipulation). There is no visible control.

Type	**Principle**	**Explanation**
Problems	1. Visibility	The only visible element is the cursor, which does not provide indications of system status, context, or how to interact with the interface.
	2. Internal model	Whilst zooming in, the action of selecting the region to see the detail makes sense. A matching action does not exist for zooming out.
	6. Recognition	The tool does not follow any real world metaphor for zooming in or out, which will require recall.
	8. Minimalist design	There is no visible control, so no information is provided other than that in the view. This will not provide a sense of position to help maintain context.
Warnings	3. User control	The interaction will need to be carefully designed so that the user can achieve what they want in a controlled manner (e.g. if the user wants to zoom in a small amount rather than a large amount).

On-view Slider

The On-view Slider, as provided by Google Maps, is a slider that is displayed over the view that it controls (see Section 2.2.1.2 Sliders).

Type	**Principle**	**Explanation**
Problems	1. Visibility	As the control is displayed over the view, it will obstruct a part of the view.
	2. Internal model	Zooming does not have a directional response in the same way that panning does so an orientation cannot be logically assigned.
Warnings	3. User control	The number of values that can be selected on a slider is limited to the discrete number of positions on it.
	4. Standards	It is not standard practice to place a control over a view.
	6. Recognition	As it is not standard, users may not recognise the control there and will require recall to use it.
	6. Recognition	Labelling will be important for users to recognise the purpose of the control.

6.4 Discussion

From this evaluation, the designs and the number of problems and warnings identified were tabulated. The designs were then ranked, best to worst, according to the number of problems and then the number of warnings. These rankings are shown in Table 13 and Table 14.

Table 13 – Ranking of panning designs

Design	Problems	Warnings
Scroll bar	1	1
Connected view	1	4
Buttons	2	3
Direct Manipulation	4	2
Circular map	4	3

Table 14 – Ranking of zooming designs

Design	Problems	Warnings
Slider	1	2
Buttons	2	3
On-view Slider	2	4
Select Level Control	3	2
Magnifying Glass Tool	4	1
Marquee Tool	4	1
Dynamic Zoom	5	1

The designs for further investigation were then chosen according to the number of problems identified for each design. Warnings identified for each design will be taken into account in the development of the prototypes.

The panning designs selected for further investigation were Scroll Bar, Connected View, and Buttons. The zooming designs chosen for further investigation were Slider, On-view Slider, and Buttons. In addition to these designs, two other designs that may have merit in special cases were chosen for investigation. The designs to be tested in this way were the Magnifying Glass Tool and the Dynamic Zoom tool. It was decided to test the Magnifying Glass Tool because it is commonly used outside the domain, so familiarity may make it a fair candidate for novice users. It was decided to test Dynamic Zoom because, while not suitable for novice users, it may be of interest to expert users because it provides a shortcut to perform tasks.

6.5 Conclusion

Three designs for panning and three designs for zooming were selected for examination in usability testing. Two other designs that may benefit users in special cases were also selected. The next chapter discusses how these designs were produced as prototypes in preparation for usability testing.

Chapter 7: Prototypes

Eight interface designs for viewing genetic sequences were selected for further investigation in Chapter Six: three designs for panning, three designs for zooming, and two direct manipulation designs. The selected designs for panning were Buttons, Scroll Bar, and Connected View. The selected designs for zooming were Buttons, Slider, and On-view Slider. The two direct manipulation designs were Dynamic Zoom and the Magnifying Glass Tool. This chapter discusses the development of prototypes for these designs to be presented to users in a usability study (described in Chapter 8).

Prototypes were discussed in Section 2.2.2 and it was decided to make these as high-fidelity prototypes because of the focus of the research on navigation. These prototypes were produced using Macromedia Flash (Macromedia, 2004), an application for developing multimedia presentations and interfaces. Macromedia Flash was chosen for its ability to provide animation and interactivity. It has been used for prototyping in other studies (e.g. Cheng and Gruen, 2003) and has been described as an "industry standard" for high-fidelity prototypes (Wigdor, 2002).

7.1 Visualisation of Genetic Sequences

The images for the genetic sequences used in the visualisation were obtained from the application Artemis 7 (Rutherford et al., 2000), examined in Chapter Five (see Section 5.1.3). Artemis was selected as it provided a clear representation of a sequence that could easily be utilised for the prototypes. In Artemis, a genetic sequence is displayed along the horizontal axis with locations of features displayed in rows parallel to the sequence (see Figure 20). This follows the representation for genetic sequences described by Lorraine and Helt (2002) (outlined in Section 2.6.1) and was common amongst the applications examined in Chapter Five.

Two example screenshots from Artemis that were used to produce the visualisation prototype are shown in Figure 20 and Figure 21. The same "repeat region" feature seen in the detail view (Figure 20) between bases 503-526 can be seen in the overview (Figure 21), illustrating the method of overview. For simplicity, later example sequence images in this chapter will show the two middle tracks of the sequence only.

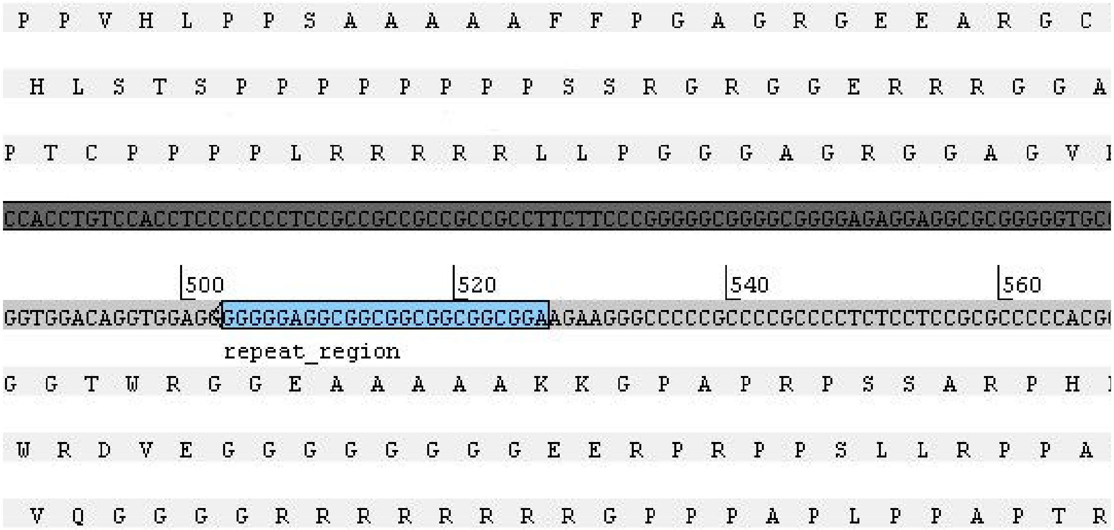


Figure 20 – A screen shot from Artemis used to produce the highest level of magnification, the detail view, in the visualisation prototype

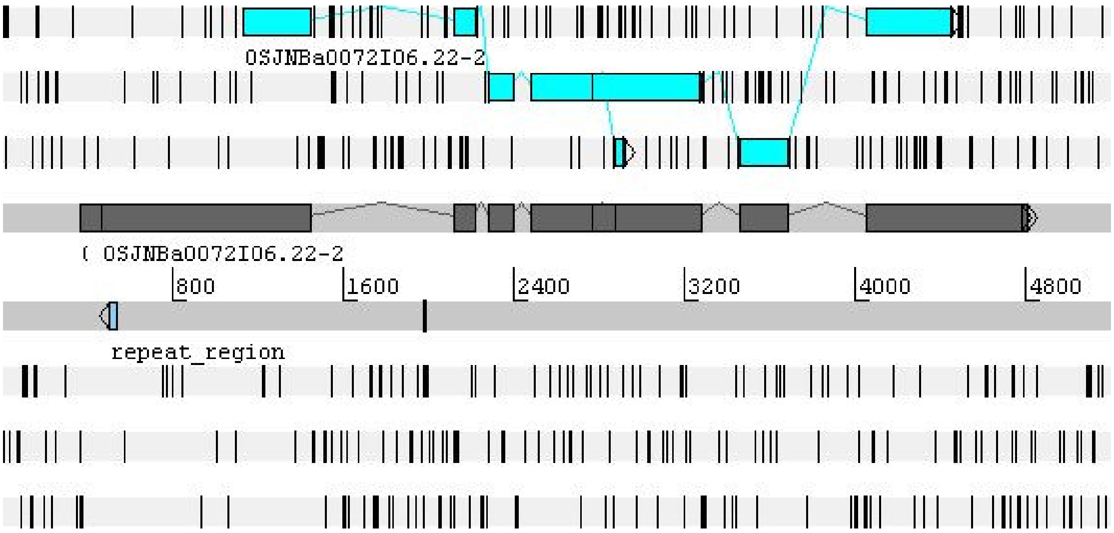


Figure 21 – A screen shot from Artemis used to produce the lowest level of magnification, an overview, in the visualisation prototype

Transitions of the visualisation for panning and zooming were animated to give a smooth and continuous appearance. The advantages of animation in aiding navigation through information spaces is well supported (discussed in Section 2.3.5) although this aspect has not been included in existing bioinformatics applications. Less than half of the applications examined in Chapter Five provided smooth transitions for panning. Zooming, if offered at all, was abrupt in all but one application and even in that application the transition was not smooth.

7.1.1 Creating the Sequence

Five different sequences of similar sizes (4000-5000 base-letters) were produced as Macromedia Flash movie objects that could be panned and zoomed. The sequences were selected to be visually interesting, containing many features at various levels of magnification. Each sequence visualisation was created by taking successive screenshots for the length of the sequence displayed from Artemis. These screenshots were edited in Corel Photopaint 9™ (Corel Corporation, 2003) to trim the surrounding screen display and isolate the sequence, and then joined together within Macromedia Flash. The joins were made as seamless as possible, so the user would not be aware of them. The entire Detail View of the typical sequence comprised thirty-two images, a total width of 27,224 pixels.

This was a simple method by which to create the full sequence. However, the images became blurry as quality was lost because Macromedia Flash stored the images internally as low quality JPEGs. This could not be circumvented, but was felt not to be a significant issue for the purposes of this project.

The sequences were then prepared to be panned and zoomed for display and navigation within the usability study application.

7.1.2 Panning the Sequence

As the sequence was now a long image (wider than the view) a mask was used to show only the region of the sequence that was within the view (see Figure 22).

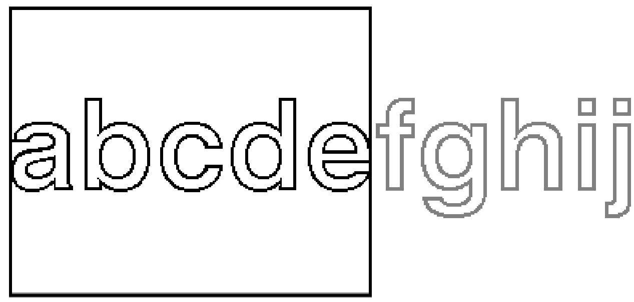


Figure 22 – Showing how the view was created. The grey part of the sequence was outside of the view and therefore not visible.

Panning was produced by moving the sequence behind the mask (see Figure 23), performed by moving the sequence along the horizontal axis. The effect of this was that the sequence could be panned horizontally and the motion of this action was animated to occur continuously.

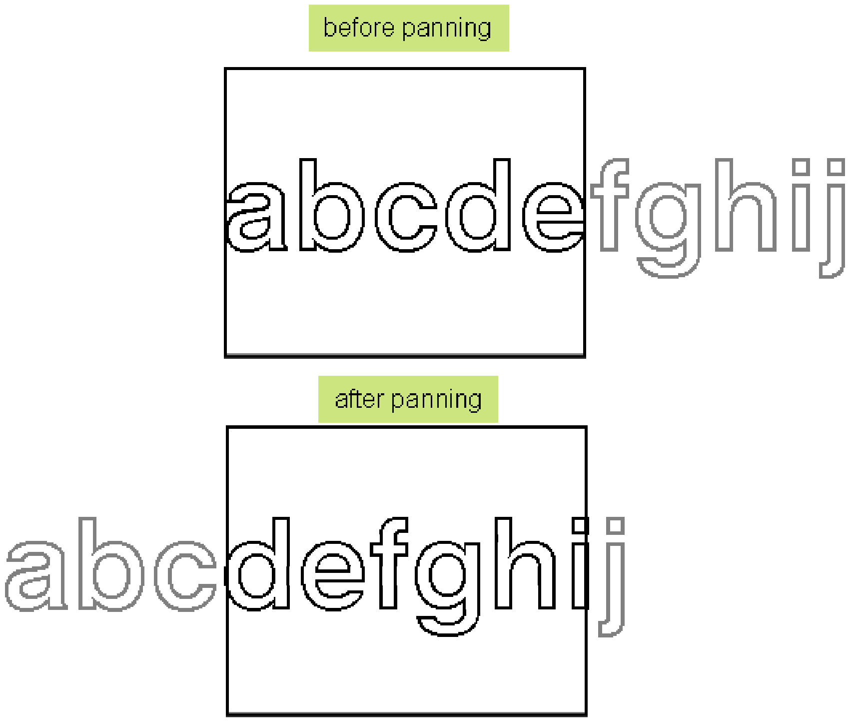


Figure 23 – Showing how panning was performed. Note that the sequence moves while the view is fixed in place.

7.1.3 Zooming the Sequence

More detail is shown at higher levels of magnification, with the bases visible at the highest level of magnification. The effect of zooming in or zooming out was achieved by changing to an image at a different level of magnification. Within Artemis a sequence can be displayed at different levels of magnification, so complete images of the sequence at differing levels were created as described in Section 7.1.1 and illustrated in Figure 24. To zoom in or zoom out simply required replacing the current image with another at a higher or lower level of magnification.

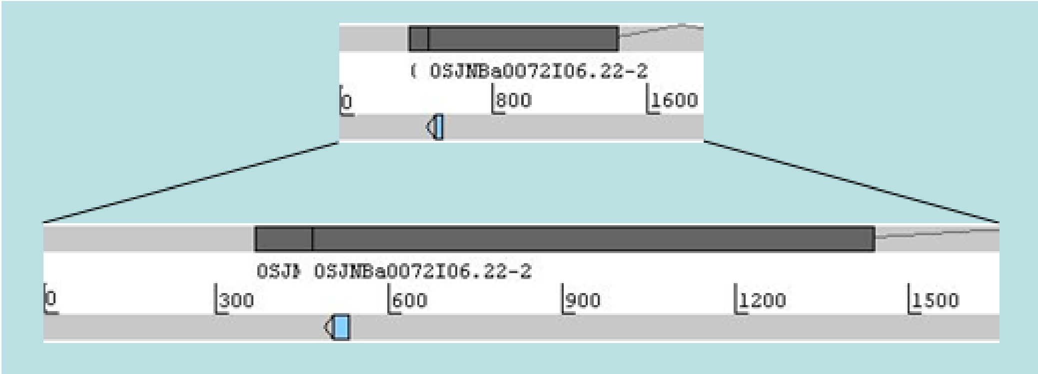


Figure 24 – Showing two different levels of magnification for the same region of sequence (0 to 1600 base pairs)

The levels of detail available from Artemis were jerky with large changes between levels of magnification. Using just these levels would not provide a smooth zooming effect. To improve the effect, as an image is zoomed it is stretched or compressed to give the illusion of more or less detail. When the zoom reaches the detail level of another image, the new image replaces the previous image in the display. The new image may also be displayed compressed or stretched to try to make the transition as seamless as possible (Figure 25).

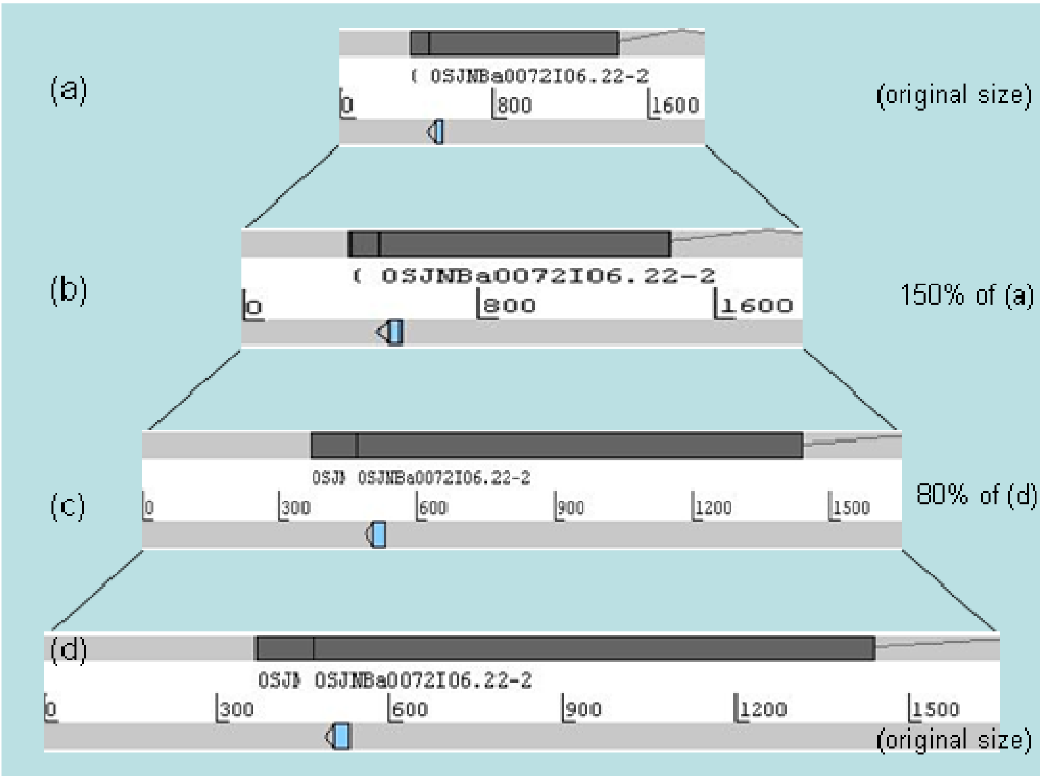


Figure 25 – Showing how compressing and stretching was used to create smooth transitions for magnification (a) first image normal size, (b) first image stretched, (c) second image compressed (d) second image normal size.

The effect of this was that the sequence could be continuously zoomed in from detail-level to overview-level smoothly.

7.2 Prototypes for Panning

Three designs were selected to be used as prototypes for panning. These designs were Panning Buttons, Scroll Bar, and the Connected View.

The orientation of the prototypes for panning was horizontal to match the orientation of the sequence in the visualisation, which in itself resolved most of the internal model issues identified in the heuristic evaluation. The prototypes for Buttons and Scroll Bar were displayed beneath the visualisation, and the prototype for Connected View was above the visualisation (as it was in the applications examined in Chapter Five).

7.2.1 Panning Buttons

The issues identified in the heuristic evaluation for Panning Buttons were lack of visibility of system status and limitations to user control. It is also important that the design match the user's internal model and that the labelling of the buttons is easily recognisable. These were considered in the development of the prototype. In the prototype, six buttons were provided grouped together horizontally, as shown in Figure 26.

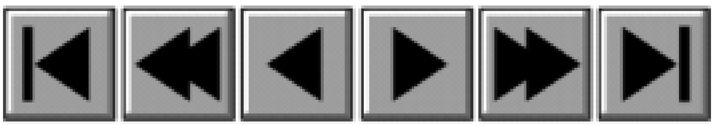


Figure 26 – The Panning Buttons prototype

The buttons were labelled with arrow heads pointing in the direction of the action that they performed. This conveyed the purpose of the button simply and corresponds to other common designs. The action that each button performed when clicked is described in Table 15.

Buttons of different speeds, similar to 'fast forward' and 'rewind', were provided to increase user control. These buttons had a double arrow head to convey a faster speed. In addition, buttons to go directly to the start and to the end of the sequence were included. These buttons were added because the symbols are generally well understood, were potentially useful and reinforced the meaning of the other buttons.

Table 15 – Showing the action performed by each of the buttons

Button	**Action**
	Pans the sequence to the start
	Pans the sequence to the left
	Pans the sequence to the left at twice the speed of previous button
	Pans the sequence to the end
	Pans the sequence to the right
	Pans the sequence to the right at twice the speed of the previous button

The buttons could be clicked to perform their action, however repeatedly clicking the buttons to move a long distance would be tedious. For this reason, another method of interaction was added, whereby pressing and holding the button repeats the action. This behaviour is common in other button interfaces, for example the buttons on scroll bars. It was considered that the longer the user held the button the further they were likely to want to move, and so the movement was accelerated the longer the button was held. The speed did not increase beyond the user's ability to see features of the sequence.

This prototype includes behaviours that were not considered during the heuristic evaluation and so the conclusions made in that section may not apply. However, it was felt that the inclusion of these behaviours alleviated some of the issues raised, particularly improving the development of an internal model, recognition and user control. Ability to show system status in the display was not resolved.

7.2.2 Scroll Bar

The issues identified in the heuristic evaluation for the Scroll Bar design were limited user control and the need to match the user's internal model. Scroll bars are a common user interface element so the prototype was made to resemble scroll bars found in other applications (Figure 27). To maximise the range of values that the track represented, the Scroll Bar was made as wide as possible, with the intent of increasing user control. The horizontal orientation of the control will match the user's internal model, resolving that issue.

Figure 27 – The Scroll Bar prototype

Clicking on the arrow buttons at either end of the track panned the sequence in that direction by a fixed amount. Clicking on the track either side of the thumb panned the sequence by a half-view in that direction. Dragging the thumb along the track continuously panned the sequence. This does not resolve the issue of limited user control, but attempts to alleviate it.

7.2.3 Connected View

The issues identified in the heuristic evaluation for Connected Views were limitations to user control and that the Connected View is not a standard user interface component. It is also important that the design matches the user's internal model, the design interaction be easily recognisable and that the design not provide too much information. There are many similarities between this design and the Scroll Bar design so the Connected View prototype was based on the Scroll Bar prototype. The similarities between the Connected View and Scroll Bar should make interacting with the Connected View recognisable and predictable. The Scroll Bar track is overlayed with an overview of the sequence, and the thumb is replaced with a 'window', or 'viewfinder' (see Figure 28). To maximise the range of values that the track represented, the overview was made as wide as possible with the intent of increasing user control. The horizontal orientation of the control will match the user's internal model, resolving that issue.

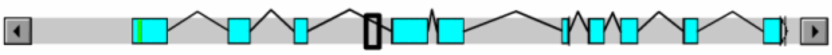


Figure 28 – The Connected View prototype

The overview was made to show the larger features of the sequence, constructed using the complete overview from Artemis. Clicking on the button at either end of the Connected View panned the sequence in that direction by a fixed amount. Clicking on a location in the overview moved the view to that location. This interaction was added as users are likely to want to "go to" a particular feature and should increase user control. Panning by this last method was performed by a smooth transition from the start point to the new location to help the user maintain context.

7.3 Prototypes for Zooming

Three designs were selected to be used as prototypes for zooming. These designs were Zoom Buttons, Zoom Slider, and an On-view Slider.

The orientation of the prototypes for zooming were made vertical, to distinguish these actions from panning. The issue identified in heuristic analysis, that zooming does not have a directional response, could not be logically resolved with these prototypes but will be investigated in the usability study. The zooming prototypes were displayed to the right of the visualisation. Each prototype could be used to zoom the sequence from detail-level to overview-level.

7.3.1 Zoom Buttons

The issues indentified in the heuristic evaluation for this design were lack of visibility of system status, lack of directional model for zooming and limitations to user control. It is also important that the labelling of the buttons is easily recognisable. These were considered in the development of the prototype. In the prototype, two buttons were provided grouped together vertically, as shown in Figure 29. The buttons were for the actions of 'zoom in' and 'zoom out' and labelled with the symbols '+' and '–'. User acceptance of these symbols will be investigated in a separate section of the usability study.

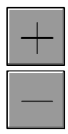


Figure 29 – The Zoom Buttons prototype

Buttons could be pressed and released to perform the zooming action. Pressing and holding the button continued the action at a constant speed. Buttons of differing speeds were not provided as there was less of a distance between the lowest and highest values for magnification.

7.3.2 Zoom Slider

The issues identified in the heuristic evaluation for this design were the lack of a directional model for zooming and limitations to user control. It is also important that the labelling of the design is easily recognisable. The Zoom Slider prototype was based on the Scroll Bar prototype. The track and thumb components were modified to resemble components of sliders found in other applications. The Zoom Slider is interacted with by dragging the thumb up or down the track. In addition, buttons were provided at the top and bottom of the slider to incrementally zoom in and zoom out (see Figure 30). These buttons were provided to increase user control and behaved as for the previous prototype.

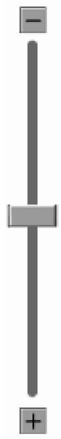
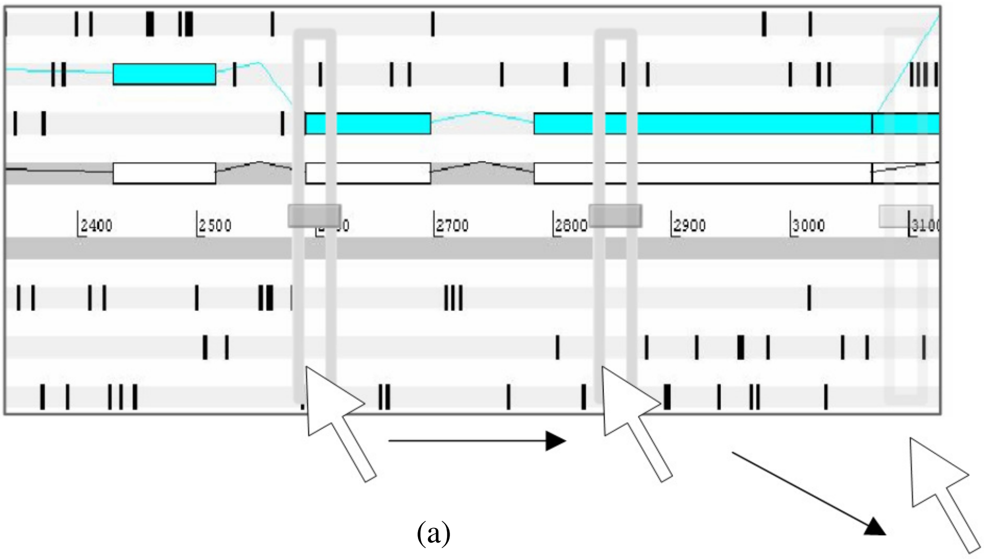


Figure 30 – The Zoom Slider prototype

7.3.3 On-view Slider

The issues identified in the heuristic evaluation for the On-view Slider design were the visibility of the design, lack of a directional model for zooming and limitations to user control, and that it is not standard to place a control over a view. It is also important that the location and purpose of the design be easily recognisable.

This prototype was based on the Zoom Slider prototype described above but with the slider displayed over the view (see Figure 31). Instead of a single track, two vertical lines were used (see Figure 31). This was intended to draw more attention to the control without obstructing a larger area of the sequence. Interaction with the On-view Slider is as for the Zoom Slider, i.e. the slider could be interacted with by dragging the thumb up or down the track. Instead of having a fixed location, the prototype follows the horizontal mouse position over the view (see Figure 31a). This was implemented to reduce the intrusion of the control on the view, particularly when users want to view the beginning or end of the sequence. When the mouse is not over the view, the control is made semi-transparent to reduce the issue of visibility (see Figure 31b). The '+' and '–' buttons of the previous design were not included which would have further impeded visibility.



(a)

(b)

Figure 31 – Showing the behaviour of the On-View Slider as (a) the mouse moves over the view, and (b) the mouse moves outside the view. Note: this is a re-creation and mouse cursor size is exaggerated.

A behaviour referred to in this research as 'side-panning' was implemented in this prototype. This behaviour panned the view left or right when the mouse was placed over the edge of the corresponding side of the view. This was expected to give more user control for moving small distances and browsing. The issue of this design not being standard cannot be addressed in the prototype.

7.4 Other Prototypes

Two other interaction designs were also selected in the previous chapter for investigation. These were Dynamic Zoom and Magnifying Glass Tool.

These designs have no visible controls and instead allow the user to directly interact with the view. It was considered that these interaction styles could be useful to users in some circumstances.

7.4.1 Dynamic Zoom

Dynamic Zoom is a mode of interaction that allows the user to pan and zoom the view by direct manipulation with the mouse. Usually this is by clicking on the view and dragging up to zoom in or dragging down to zoom out (Sahlin, 2003).

The issues identified in the heuristic evaluation for Dynamic Zoom were lack of visibility, lack of directional model for zooming, that the design is not yet standard and there are no cues on how to interact with the interface. It is also important that care be taken in implementation to provide user control.

Zooming and panning are often required together to locate particular areas of a sequence. In this design, zooming is done by dragging vertically while panning is performed by dragging horizontally. It is possible to perform both actions in one movement by dragging diagonally. To prevent small movements from the user being interpreted as actions, a 'buffer zone' was used (see Figure 32). This meant that the user had to drag the mouse a minimum distance in either direction before the corresponding action would be performed (see Figure 33).

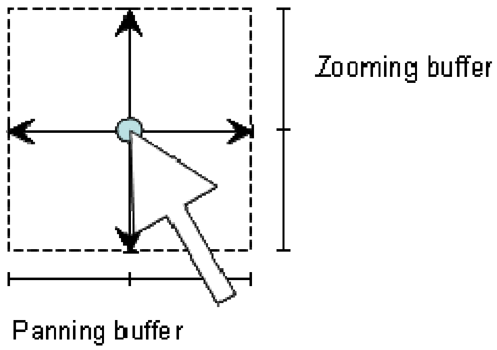


Figure 32 – The Dynamic Zoom tool required the mouse be moved outside a buffer zone before panning or zooming was initiated

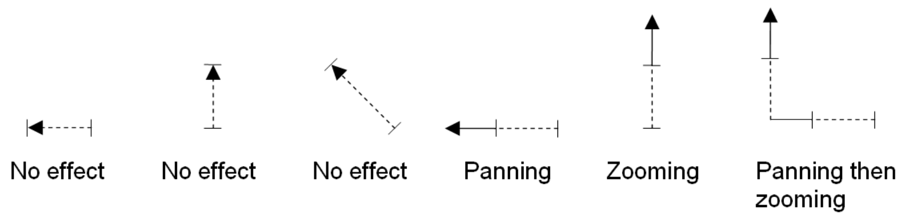


Figure 33 – Showing the effect of different actions using dynamic zooming. Note: Dragging in the opposite direction had the opposite effect. Dashed lines indicate movement within the buffer zone.

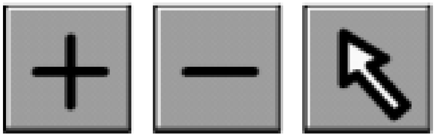
The sequence continued to pan while the mouse button was depressed on the view, and the further the mouse was dragged, the faster the sequence panned or zoomed. These behaviours should improve user control.

No visual indication of the mouse mode or buffer zone was provided, and that no control or feedback on system status is visible is inherent to this design. This also means that there is no external model of how the design operates and so the user will have to learn how to use it.

7.4.2 Magnifying Glass Tool

The Magnifying Glass Tool is a mode of interaction that allows the user to click directly on the view to zoom in or zoom out, centring the view on the point clicked. The issues identified in the heuristic evaluation for the Magnifying Glass Tool design were lack of visibility and the incongruence of the metaphor for zooming out. It is also important that the mode of operation selected is easily recognisable and that care be taken in implementation to provide user control.

Buttons were provided above the view to select which mode to operate in (see Figure 34). The mouse cursor was changed to indicate the currently selected mode (see Figure 35), which will increase visibility of the system status and through use of the magnifying glass metaphor should increase recognition of the design.



(a) (b) (c)

Figure 34 – Buttons provided to select mode: (a) zoom in, (b) zoom out, (c) pan

(a) (b) (c)

Figure 35 – Mouse cursor symbols used to portray mode: (a) zoom in, (b) zoom out, (c) pan

Zooming was performed by selecting 'Zoom in' or 'Zoom out' mode and clicking on the region of the view to be zoomed on. Zooming continued until the mouse button was released. Panning was performed by selecting panning mode and dragging horizontally. The zooming or panning action continued until the mouse button was released.

That the metaphor for zooming out is incongruent is inherent to the design has not been resolved.

Chapter 8: Usability Study Method

The prototypes described in Chapter Seven were presented in a usability study to a sample of users who were observed performing tasks with the prototypes. This usability study is to directly observe the user experience of navigating sequences in applications. It will give an understanding of users' abilities and their expectations of the system and help to discover serious problems that users may encounter. This also gives the opportunity to investigate user understanding, user terminology and design orientation. The results are presented and discussed in Chapter Nine.

8.1 Trial Organisation

Recruitment

The optimal number of participants for a small usability study is seven people (Nielsen & Landauer, 1993) so it was intended to perform usability trials with five to ten participants. A recruitment letter (see Appendix C) was sent to all respondents of the initial questionnaire who indicated a willingness to be contacted for participation in further research. A general invitation was also sent to staff of the Bio-Protection and Ecology Group at Lincoln University and the Germplasm Enhancement Group at Crop & Food Research Ltd. The participants that were selected were those who had some experience of working with genetic sequences on computer.

Role of the researcher

Typically a usability trial will involve a moderator and several observers (Kuniavsky, 2003). In this study the researcher took on both of those roles. The disadvantage of this is that one person is less able to observe and record all the required information. To ease the workload of the researcher, the trial was made to be largely self-directing. The researcher sat adjacent to the participant to be able to both observe and communicate with the participant.

Trial conditions

Trials were scheduled to last for one hour. One trial only was scheduled per half-day to allow for writing up observations and finalising data collection. Trials were conducted in the researcher's office which was set up for the purpose. Throughout the trial, the tone of discussion and questions was kept informal. During the trial the researcher made notes on a purpose-designed 'observer sheet' (see Appendix G). This sheet allowed rapid data collection for many aspects of the trial. Quotes from participants were also noted where possible and relevant. The computer screen was recorded for the duration of each trial using software called Camtasia (TechSmith).

Human Ethics Committee (HEC) approval was obtained before commencing the study (Application #HEC 2004-31).

Trial Tasks

The tasks that users performed during the trial were chosen to be simplified representations of common tasks that may be performed by users based upon experience and the information collected in the initial questionnaire. These tasks were also to be representative of the three navigational activities described in Section 2.3.4: wayfinding, exploration, and identifying objects.

The objects involved in each task were identified by their technical description (e.g. "exon"), and a generic term (e.g. "light blue box"). The technical term allowed the participant to think in terms of a real project whilst the generic description meant no person was disadvantaged. If the participant did not understand the terminology, the researcher explained it.

Three tasks were given to participants for each design, as described in Table 16.

Table 16 – Description of tasks and examples of actual questions

Activity	**Task**	**Example of question**
Wayfinding	To browse to reach a specific location	"What is the location of the first base (letter) of the first exon (light blue box)?"
Exploration	To navigate to a location	"What are the first four bases (letters) on the reverse sequence from location 2000?"
Identifying objects	To determine information about a generalised group	"How many boxes are there between the first exon and position 2000?"

The same tasks were tested for each design, changing the target for each design. Tasks were revealed sequentially and each task had to be completed before the next was revealed. The intention of this was to focus the participant on the current task.

A more informal approach was taken with the Dynamic Zoom and Magnifying Glass Tool designs. These designs were thought to have merit in some special cases, but they did not warrant the time required for a full evaluation. Rather than performing a complete usability test for these designs, participants were presented with the design and asked to experiment with it, without being given specific tasks to perform.

8.2 Trial Process

When the participant arrived, they were seated at the computer desk and a brief casual conversation was allowed to occur. This allowed the researcher to introduce himself, develop rapport and let the participant relax. When the participant was comfortable, the trial began.

The first section of the Usability Script (see Appendix E) was read to the participant, explaining the purpose of the study and outlining the conditions of the trial. Participants were asked to sign a statement of informed consent (see Appendix D).

The script explained to the participant that what they were being shown were prototypes of possible interfaces and it was stressed that there was no commitment to these prototypes or designs. This was emphasised to reduce the participant's inhibitions so that they felt free to make criticisms because they knew that their feedback would save development time. Participant's responses would then be less biased and more useful.

The trial had five sections (each was called a "study" in the trial, as shown in Figure 36). Each section had a different focus:

- Section One investigated the terminology of the participant
- Section Two tested usability of designs for panning
- Section Three investigated the meanings of symbol and orientation to participants
- Section Four tested usability of designs for zooming with panning
- Section Five briefly investigated usability of interfaces that may benefit users in special cases.

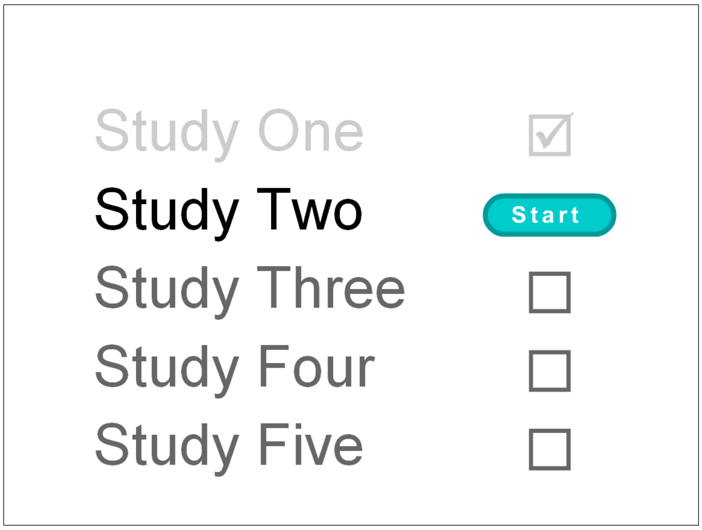


Figure 36 – Screen showing trial progress

To lead the participant through the trial, a screen showing trial progress was used. This screen displayed the list of sections to be completed. The next section to be performed was displayed with a 'Start' button; completed sections were displayed dimmed and with a ticked box; and incomplete sections were displayed with an empty check box (see Figure 36, above).

Each section was preceded by a screen describing the purpose of the section and concluded with a screen explaining the importance of the information collected. Between sections the screen showing trial progress was displayed.

The participant was given some control of the trial progress by the provision of a button for them to click when they were ready to advance. However, the button to advance was not displayed until an event occurred: the passing of a time period or the participant interacting with the prototype. This was to ensure that each step was followed and prevented the participant accidentally clicking the button twice or seeing a button and clicking instinctively.

Most of the trial was spent observing the participant's use of the prototypes. A brief satisfaction questionnaire was filled in at the end of each section.

8.2.1 Section One: Terminology

To enhance communication with the user, it is important that any resultant application uses terminology familiar to the users. This section was used to collect terminology from participants. To avoid the participant being biased by terms used in the study, this section was performed at the start of the trial.

Participants were shown a view of a genetic sequence and asked if they recognised it as a genetic sequence. Then a section of the view was highlighted and the participant was asked to give their term for what was displayed (see Figure 37).

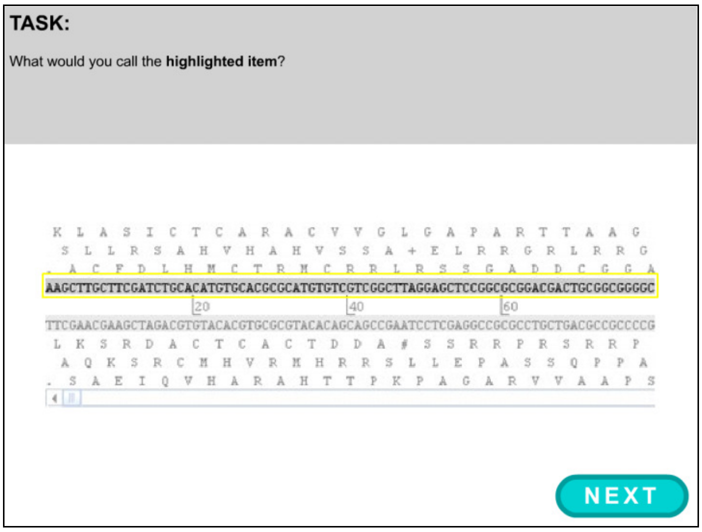


Figure 37 – An example of how concepts were presented to participants, highlighting the “forward sequence”.

The participant’s response was collected verbally, and the next item shown. Eight concepts were investigated:

1. Forward sequence
2. Reverse sequence (or how this is distinguished from #1)
3. Base number, location or position
4. Detailed view or sequence view
5. Panning (demonstrated by animation)
6. Zooming (demonstrated by animation)
7. Overview
8. Features or annotations.

This section was also important as it encouraged the participant to talk about what they were seeing and thinking, which was required for later sections.

8.2.2 Section Two: Usability Test with Designs for Panning

Three designs for panning were investigated: Panning Buttons, the Scroll Bar, and the Connected View. The same sequence was used for each design in the section.

First, the design was displayed (see Figure 38) and the participant given time to familiarise themselves with the design. The participant's first impressions were elicited and recorded by the researcher. The participant was asked about their terminology for the design, whether they recognised the design and how they thought that they would interact with the design.

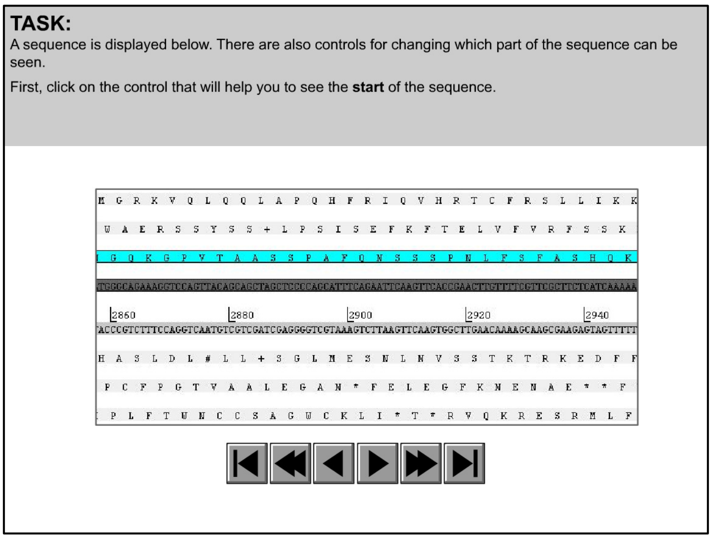


Figure 38 – An example of how the designs were presented to participants, showing the panning buttons

The views were initially centred each time, so that Scroll Bar and Connected View which portrayed a range of values were also initially midrange. This was so that the position did not give the cue to the meaning of the designs, rather than its appearance and labelling.

Next, the participant was asked to perform three tasks using the design (see Figure 39).

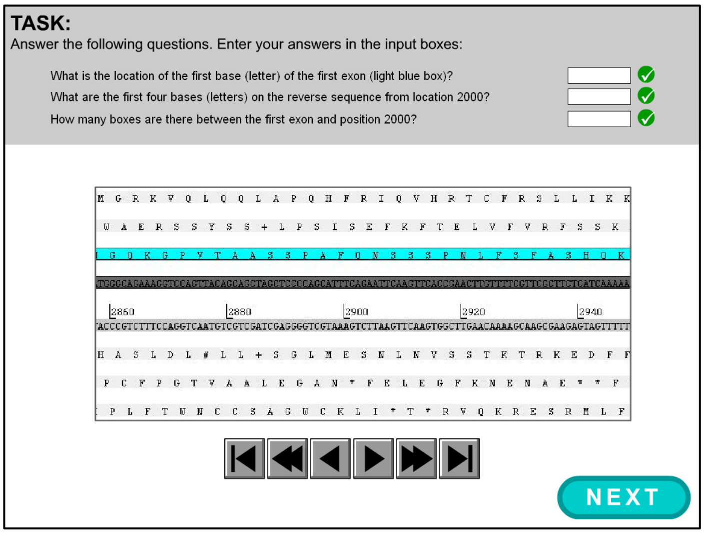


Figure 39 – Showing how tasks were presented to participants

After the tasks were completed, the participant's experience with the design was discussed. The participant was asked how well the design performed, how suitable it was for the tasks and how it could be improved. When discussion of the design was finished, the participant was shown the next design.

At the end of the section, the participant was shown the three designs that they had just used (see Figure 40) and asked to complete a 'Task Satisfaction Questionnaire' (see Appendix F). The Task Satisfaction Questionnaire asked the participant what their preferred design was for each of the tasks, what their preferred design was overall, for further comments and suggestions on the designs investigated, and for suggestions on other methods of performing the same action.

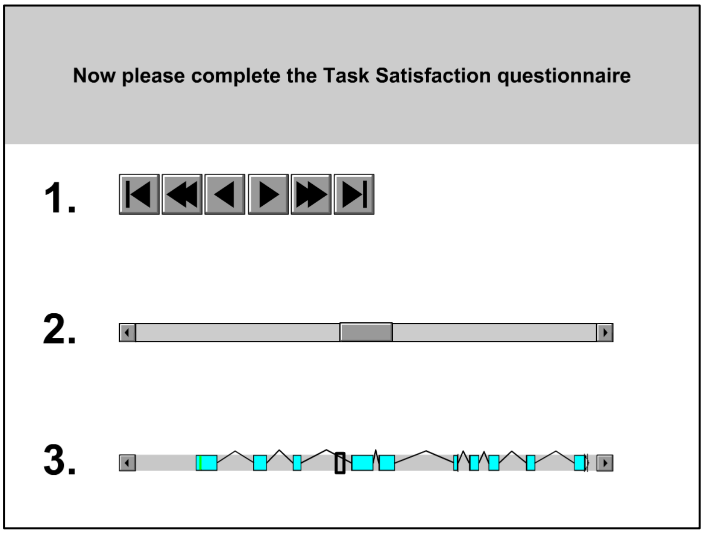


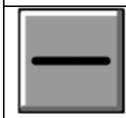
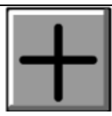
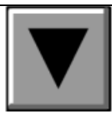
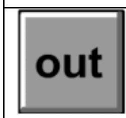
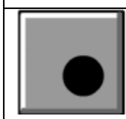
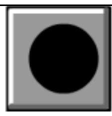
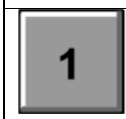
Figure 40 – Showing how the designs were presented at the end of the section

8.2.3 Section Three: Symbol and Orientation Meanings

This section investigates a design issue that was identified as a warning in heuristic evaluation for the zooming designs. The warning was that the labelling of designs must be meaningful to users, particularly for zooming for which there is no logical orientation.

An investigation was performed in which actual buttons were displayed on screen so that participants were presented with real buttons and would have the compulsion to push a button. It was felt that seeing an active button on the screen would give the decisions more meaning and importance to participants than a static representation. The sets of symbols used in the study are shown in Table 17.

Table 17 – The symbol sets used in the study

Left	**Right**
	
	
	
	
	

First, the participant was shown two symbols that were placed on two buttons arranged horizontally (see Figure 41). The participant was asked to click on the button that most means “Show more detail” to them. When the participant made their selection they were shown the next set of two symbols. This was repeated for the five sets of symbols.

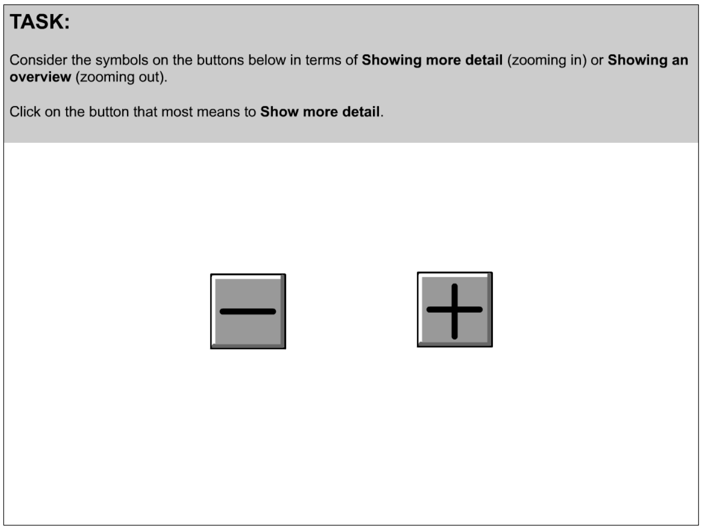


Figure 41 – Example screen for investigating meaning of symbols for the purpose of zooming

Next, the participant was shown two unlabelled buttons arranged vertically and was asked to click on the button that most means "Show more detail" to them. The buttons were unlabelled so that participants would not be biased by labelling and could act on intuition (see Figure 42).

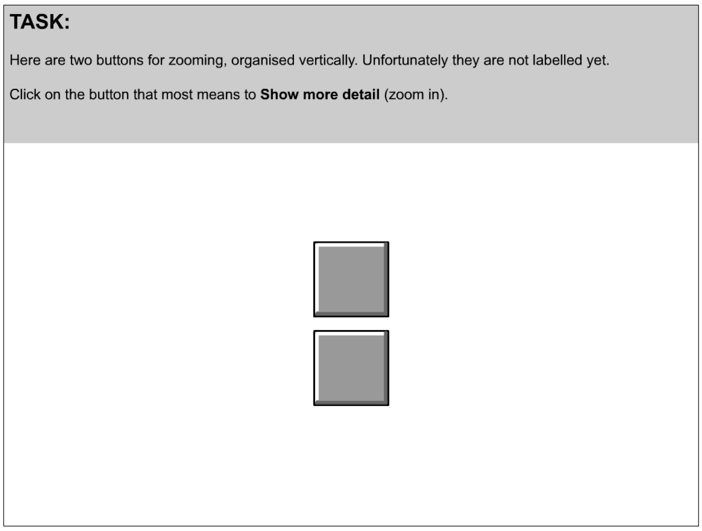


Figure 42 – Showing the screen for investigating orientation of buttons for zooming

The participant was then presented with all five symbol sets from Table 17 as sets of buttons. Within each set, the buttons were arranged so that the preferred symbol for "Show more detail" was displayed at the preferred position (top or bottom) (see Figure 43).

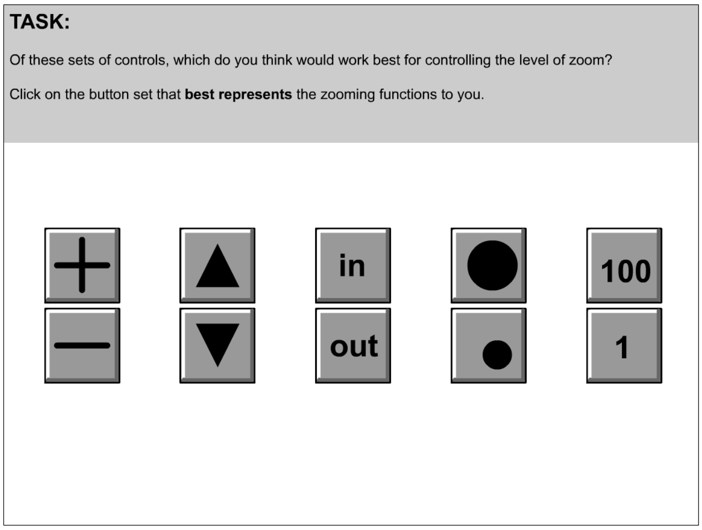


Figure 43 – Example of how button sets were presented to participants

The participants were then asked to rank the symbol sets from "best represents" to "least represents" by clicking on the sets in their order of preference. As each set was selected it was dimmed so that the participant knew that it had been selected (see Figure 44). After three sets had been selected, participants were able to progress if they could not decide between the last two.

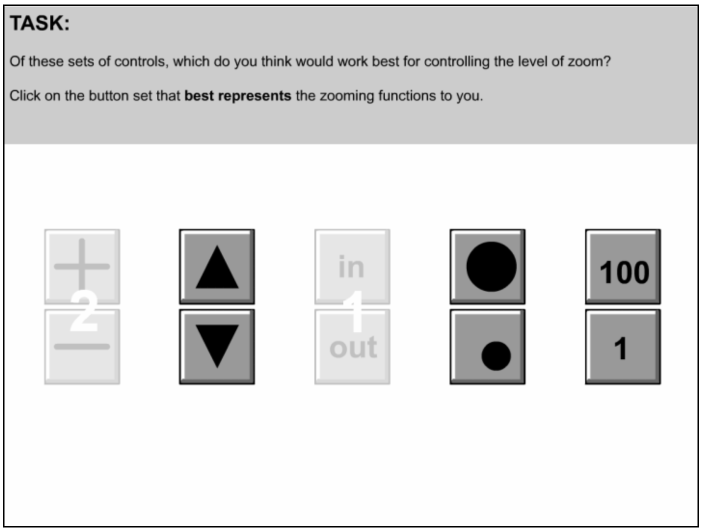


Figure 44 – Example of how button sets were presented to participants once selected

8.2.4 Section Four: Usability Test with Designs for Zooming

The same basic procedure was used for this section as for Section Two. Three designs for zooming were investigated: Zooming Buttons, Zoom Slider, and On-view Slider. As panning was also required for these tasks, a Scroll Bar was provided. The Scroll Bar was chosen because it is the most predictable and recognisable tool of those examined so participants were more likely to be able to just use it rather than have to think about it. The impact of this decision was not tested. The same sequence was used for each design in the section.

At the end of the section, the participant was shown the three designs that they had just used (see Figure 45), and asked to complete a 'Task Satisfaction Questionnaire' for that section.

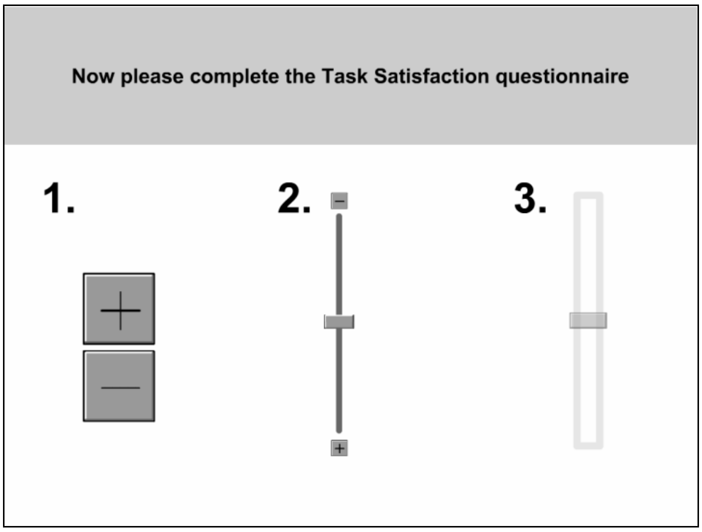


Figure 45 – Showing how the designs were presented at the end of the section

8.2.5 Section Five: Brief Investigation of Other Interfaces

Two designs were informally investigated: the Magnifying Glass Tool and the Dynamic Zoom Tool. Participants were presented with each design individually and encouraged to experiment with the design (see Figure 46) with no set tasks to perform. The participant was asked how well they felt the design performed and when they thought it might be suitable.

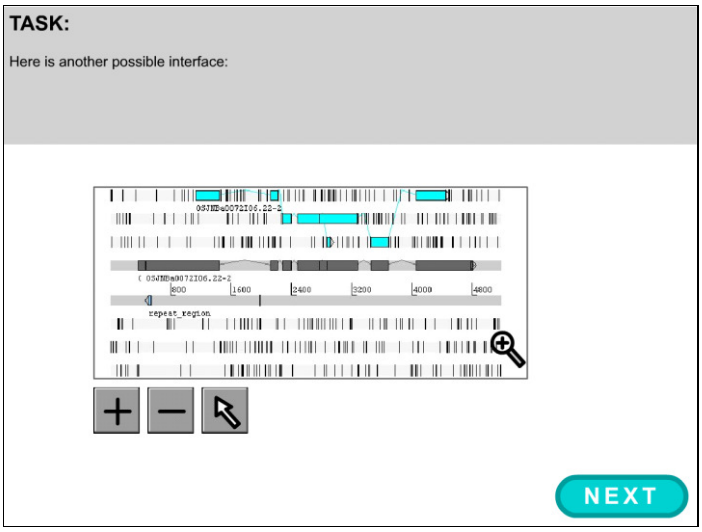


Figure 46 – Showing how the Magnifying Glass Tool prototype was presented to participants

8.3 Data Collection

Data was collected by the usability study application, by the observer, by paper forms the participant completed during the trial, and recording the computer screen.

The prototypes were programmed to record data during their operation, as suggested by Rudd, Stern and Isensee (1996). All user interface components that users interacted with were programmed to record mouse actions and associated data. The response for each task performed was also recorded. The data collected by the application for each mouse event during each trial was:

- Trial Number, uniquely identifying the trial taking place
- Task, uniquely identifying the section of the trial the user was acting in
- Event, the event being reported, 'press' or 'release'
- Element being interacted with
- Time the event occurred.

The computer screen was recorded for the duration of each trial using Camtasia (TechSmith). Camtasia was set up to capture the entire screen, add time/date to start of video for three seconds, highlight the mouse cursor with a yellow circle in the recorded video and display mouse clicks with red rings in the recorded video. Audio was not recorded.

The researcher made notes on a purpose-designed ‘observer sheet’ (see Appendix G) during the trial. This sheet allowed rapid data collection for many aspects of the trial. Quotes from participants were also noted where possible and relevant.

8.4 Pre-testing

During the development of the study, pre-testing was performed with staff members of the Applied Computing Group, Lincoln University. A trial was performed for each pre-test in the manner of a trial with an actual participant. These trials gave feedback on the style and procedure of the study from a participant’s perspective and helped refine the process.

During pre-testing it was noticed that:

1. Participants sometimes did not understand what was expected of them.
2. Participants did not respond well to sudden changes on-screen (e.g. when the participant has entered their response in the input box, the next task appears) or in the process (e.g. the transition from panning designs to zooming designs). Where changes were sudden, participants were either surprised, or did not notice the change.
3. Participants wanted feedback on the accuracy of their responses.
4. The data collected via paper forms was design oriented, whereas the purpose of the study needed to be task focussed.

To resolve these issues:

1. Task descriptions were made more instructive, for example, adding the statement “Enter your answers in the input box”.
2. A pause was placed between when events were triggered and the response occurred. Also, a break was placed between each section.
3. Feedback was given to participant responses in the form of a ‘tick’ being displayed when the expected number of characters was entered in the input box.

As the accuracy of responses was not an objective, responses were not checked during the trial.

4. The paper forms were reframed from being design-focussed to task-focussed.

Also, the observer sheet was refined to ease data collection. This highlights the value of both pre-testing and testing with users, as many of these problems would not have been found by the designer through inspection.

8.5 Analysis

As discussed in Section 8.3, a range of data will be collected during the trials and will be examined for indicators of effectiveness, efficiency and user satisfaction, and also for general trends and outliers. Holleran (1991, cited in Riihiaho, 2000) recommends that "designers gather multiple types of data about the usage of a product, because agreement and consistency among results enhances the probability of their validity and generality". As only a small number of participants are to participate in the study, statistical analysis will not be performed. Instead, tallying and averaging of results will be used.

The data collected from mouse events (mouse button clicks and releases) will be used to determine performance measures including the number of clicks each participant used to complete a task and the amount of time the participant took to complete a task.

The intention was to perform metric analysis, as described in Section 2.5.5.1, however it was felt that those factors contributing to metric scores were more usefully described verbally than summarised as a number.

For the purpose of judging correctness of the answers that participants give to tasks, those answers will be checked according to the following guidelines:

- For "Find Feature" tasks the expected response is a four-digit number. Responses will be accepted up to two base pairs either side of the expected response.
- For "Go to a Position" tasks the expected response is a four-letter string. Participants may read the letters left-to-right or right-to-left, depending upon interpretation. These variations will be accepted.
- For "Identify Features" tasks the expected response is a single-digit number, being the number of features between a location and specified feature. Participants may

interpret features partially within the range as included or not, and may include the specified feature, or not. These variations will also be accepted.

Those responses not falling into the accepted range will be investigated and the related video reviewed to try and understand what happened. The level of error will then be graded as “trivial” or “serious”.

Observations noted by the researcher and reported by the participant on paper forms will be examined for indicators of the participant’s ability to understand the designs and their views on the effectiveness, efficiency and satisfaction in using the designs.

Chapter 9: Results and Discussion

A sample of target users were observed in usability trials using the prototype application described in Chapter 8. This usability study allowed direct observation of the user experience of navigating sequences in applications and gave understanding of users' abilities, their expectations of the system and helped to discover serious problems that users may encounter. This also gave the opportunity to investigate user understanding, user terminology and design mapping. Each section of the study is discussed separately. An overall summary and conclusions will be given in Chapter 10.

9.1 Data Collection

Studies were performed between the dates of 15/12/2005 and 22/12/2005. Seven participants were recruited for the study. There were three male participants and four female participants. A complete trial was also performed with a participant who was not involved in preparing the usability study but was not a member of the group of target users either. This was to test the process before starting actual trials.

Issues encountered during the study were that:

- The video from one of the participants was not recorded.
- The Connected View prototype did not record events where the participant clicked on the Overview. These were reconstructed using the video.
- The panning behaviour of the On-View Slider, when placed at the left- or right-edge of the view, was not recorded.

As there were only seven participants, statistical analysis was not appropriate. Instead, a descriptive approach is used to analyse the results.

9.2 Section One: Terminology

The trial began with an examination of the terminology used by participants to describe what they worked with, ideally collecting the terms that the participants would use to describe the item to colleagues. Participants were shown a view of the visualisation on screen and asked to consider it as representing a genetic sequence. Parts of the visualisation were then highlighted and participants were asked to give their terminology for the representation.

Participants responded with a range of terms however there was strong consensus on some of them.

The first four items shown to participants are illustrated in Figure 47.

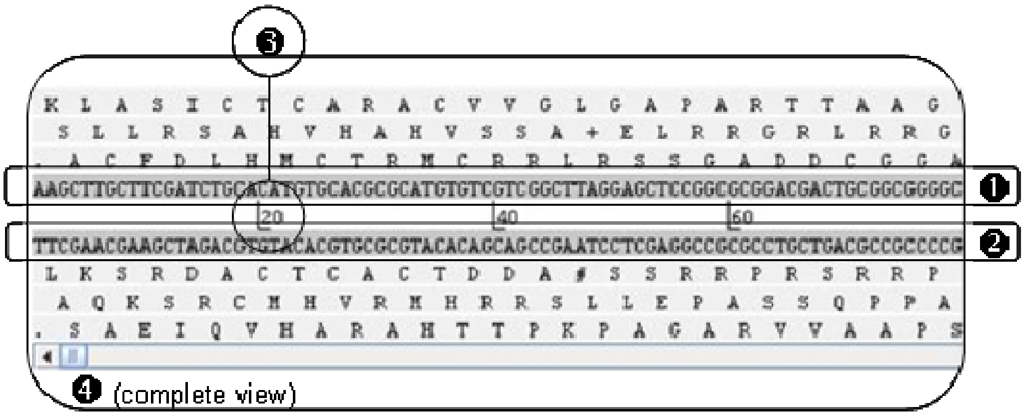


Figure 47 – Four of the items demonstrated to participants

Item 1 was referred to by three participants as “forward sequence”, by three participants as “sense strand”, and by other participants as the “positive strand” and “nucleotide sequence”.

Item 2 was referred to by three participants as “reverse sequence”, by two participants as “anti-sense strand”, and by other participants as the “negative strand” and “complementary sequence”. All participants that suggested “forward sequence” for Item 1 also suggested “reverse sequence” for Item 2. Two of the participants that suggested “sense strand” also suggested “anti-sense strand”.

Item 3 was referred to by four participants as “base number” and by three participants as “position”, and by another participant as “site”.

Item 4 was referred to by three participants as “sequence view”, and by other participants as “magnified view” and “expanded view”. Two participants gave no response.

Items 5 and 6 were actions that were demonstrated on-screen. Item 5 showed the sequence moving from side-to-side, which six participants described as “scrolling”. Item 6 showed the view being magnified in-and-out, which all seven participants described as “zooming”. There was a good common understanding for these items. The participants’ use of the word "scrolling" was interesting, contrasting with the use of "panning" throughout this research.

Items 7 and 8 that were shown to participants are illustrated in Figure 48.

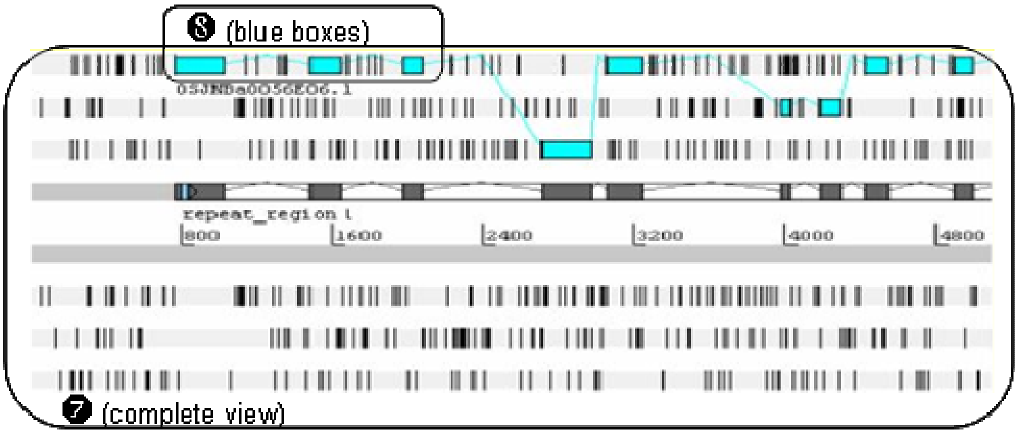


Figure 48 – Two of the terms demonstrated to participants, circled.

Item 7 was described by two participants as an "annotated view" while others described it as an "overview", "high level view" or "composite view". Two participants gave no response.

Item 8 was described by four participants as "annotations", two participants as "motifs", and by another as "areas of interest" or "features". One participant gave no response.

Participants seemed to find Section One of the study most difficult. This seemed to be due to a number of factors. There was a feeling that they were somehow 'trick questions', or participants tried to think too deeply or tried to 'out-think' the study. Another possibility was that participants had not had to name or describe those concepts before.

However, it was a good way to start the trial because, by the end of the section, participants no longer felt the study was tricky or a test of themselves and they felt free to discuss issues. This helped open up the discussion for the next sections.

9.3 Section Two: Usability Test with Designs for Panning

This section presented prototypes of designs for panning to participants and they were asked to perform tasks using those designs. These tasks, performed for each design, covered the three navigational activities described in Section 2.3.4:

1. **Find a Feature**
   For the navigational activity 'Exploration', participants were instructed to find the location of a particular feature, e.g. "What is the location of the first base (letter) of the first exon (light blue box)?"

2. **Go to a Position**
   For the navigational activity 'Wayfinding', participants were instructed to find the four bases from a position, e.g. "What are the first four bases (letters) on the reverse sequence from location 2000?"

3. **Identify Features**
   For the navigational activity 'Identify Objects', participants were instructed to browse to find out how many features existed between two points, e.g. "How many boxes are there between the first exon and position 2000?"

Participants responded well to this section, quickly establishing what they were to do and beginning to use the designs. A lot of feedback was gained during this section about each of the designs being proposed. The results for these designs will be examined together and then each design discussed individually.

9.3.1 Performance Measures

Each task in this section was successfully completed with each of the designs and none of the answers to the tasks was outside the expected range.

A comparison of the average number of clicks to complete each task is presented in Table 18 and the average time to complete each task is presented in Table 19. The time measured was from the moment the participant first used a control until they last used a control to complete the task. One participant, referred to as 'Participant A', was excluded from these results as they took significantly more time and mouse clicks to complete all tasks than other participants.

Table 18 – Average number of clicks to complete each task, excluding Participant A

	Panning Buttons	Scroll Bar	Connected View
Find Feature	7.7	4.0	1.2
Go to a Position	11.8	1.2	1.3
Identify Features	9.7	1.2	0.2

Table 19 – Average time (seconds) to complete each task, excluding Participant A

	Panning Buttons	Scroll Bar	Connected View
Find Feature	15.1	8.5	2.9
Go to a Position	14.5	4.9	5.6
Identify Features	31.8	22.9	1.6

From these comparisons it can be seen that Panning Buttons always required the most interactions to complete tasks both in terms of number of clicks and time. The Connected View required marginally more time and clicks than the Scroll Bar for the task of 'Go to a Position', otherwise it required significantly less time and fewer clicks.

Both the Scroll Bar and Connected View designs could instinctively be used by participants, while there was a small learning curve for the Panning Buttons design due to the design's behaviour. The issue of overshooting, i.e. travelling past their intended destination, affects efficiency of use. This issue was most prominent when participants were using the Panning Buttons design. It was encountered but easily corrected by participants using the Scroll Bar design, and was not observed with the Connected View design.

Participant A was not included in the comparative figures above as they took significantly more time and mouse clicks to complete the tasks compared to the other participants, except for the Connected View design which they were able to use as effectively as the other participants. This participant did not seem confident, was quite hesitant in performing actions and their typical interaction behaviour often differed from other participants. They are still relevant to this study as they were a representative user and there will be such users in the work environment. However, as their behaviour is not typical and this is a small sample they are not considered in the comparative figures.

9.3.2 Panning Buttons Design

The behaviour of the Panning Buttons design is described in Section 7.2.1 and its appearance can be seen in Figure 49.

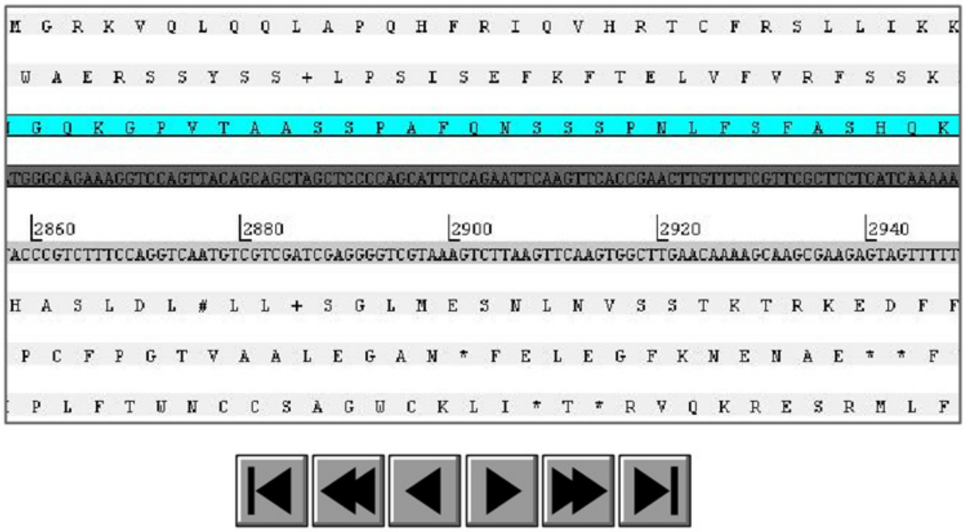


Figure 49 – Panning buttons as presented in the usability study

Three participants described this design as “scroll(ing) buttons”, while one participant described it as “buttons” and another “scrolling controls”. The design was recognised by all participants.

All participants came to understand the purpose of each button and recognised the repeating behaviour of the buttons when held down. Most participants used the buttons in this way although two participants held the buttons for relatively short periods only whilst completing the tasks. Three participants stated that learning to make controlled use of the design would take some time due to the speed and acceleration of the movements caused by the design.

The use of buttons varied by task. The ‘Pan to Start’ button was used by five participants at the beginning of the ‘Find Feature’ task; the ‘Pan to End’ button was used by only one participant. The ‘Pan Left’ and ‘Pan Right’ buttons were typically used for the ‘Find Feature’ and ‘Identify Features’ tasks and were frequently held down for two and up to seven seconds. The ‘Pan Right Faster’ was used for the ‘Find Feature’ task and particularly for the ‘Go to a Position’ task, where it was typically held down for two to three seconds. The ‘Pan Left Faster’ was used for the ‘Go to a Position’ task, where it was typically held down for short clicks only.

All participants experienced issues with ‘overshooting’. Whenever a participant overshot their target, they reverted to movement by many short clicks (see Figure 50), demonstrating learning difficulties with the design and suggesting less confidence in use.

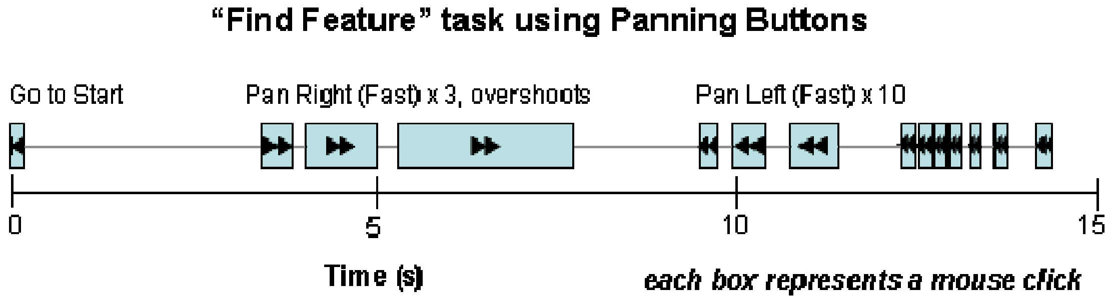


Figure 50 – Chart showing typical behaviour of a participant using the Panning Buttons design

In the example illustrated in Figure 50, the participant began by going to the start of the sequence. They then paused, perhaps to reorient themselves and ascertain whether the goal had been met (it had not). They then used the Fast Pan Right button, holding it down three times and overshooting their destination. They then clicked the Fast Pan Left button ten times to return to the target. This behaviour was typical of all users for the first two tasks. Task three, which involved scanning a region of the sequence, was generally completed by moving along the region with short clicks of the panning button.

Participant satisfaction with the Panning Buttons design was mixed, with two participants describing it as “good” but another as “annoying”. Participants particularly liked the ‘Go to Start’ and ‘Go to End’ buttons, but there were instances where participants wished they could go back to the position they had moved from. The design was described by participants as slow and did not appear to be entirely efficient or effective.

9.3.3 Scroll Bar Design

The behaviour of the Scroll Bar design is described in Section 7.2.2 and its appearance can be seen in Figure 51.

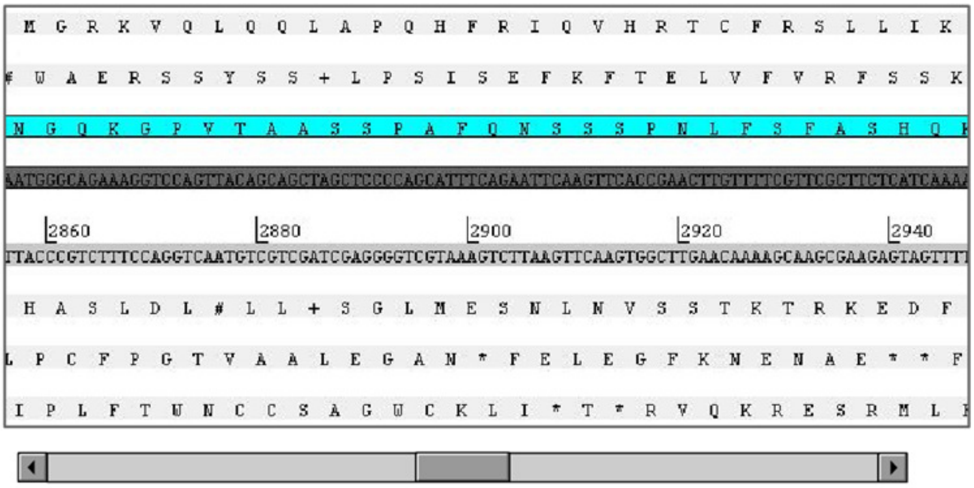


Figure 51 – Scroll bar as presented in the usability study

This design was familiar to all participants, all understood how to interact with it and all but one called it a "scroll bar".

Participants most often used this design by holding down the thumb until they were satisfied they were finished scrolling the view, including times that the mouse was stationary (see Figure 52). The consequence of this is that the measure of "time spent interacting with control" is inflated.

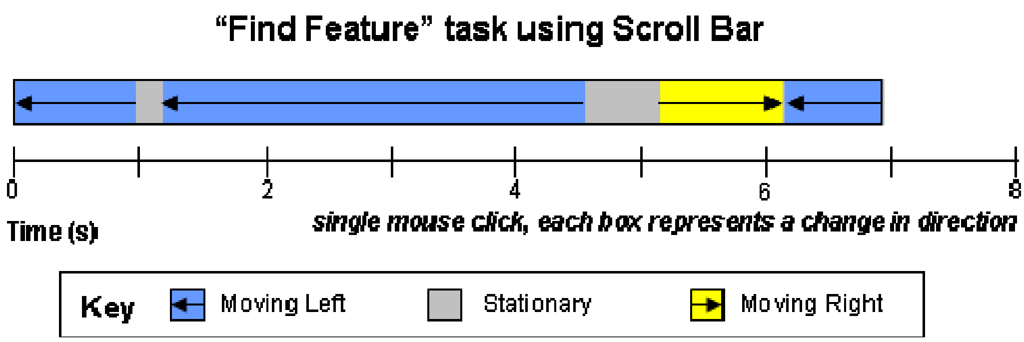


Figure 52 – Chart showing typical behaviour of a participant using the Scroll Bar design

In the example illustrated in Figure 52, the participant completed the task in one click of the mouse. They moved to the start of the sequence in two motions, pausing in between but still holding the mouse button down. This action was performed quite slowly, perhaps browsing the features of the sequence. When they reached the start of the sequence they paused again then quickly panned through the sequence to find the feature, overshooting, and then slowly panned the sequence back to the target. Participants typically moved the scroll bar slowly while browsing and quickly when they had a target to move towards.

In all but two cases, participants used the scroll bar thumb to move. In those cases, one participant used the scroll button for a task; another participant clicked in the trough to scroll. One participant stated that they would use the buttons for small movements. That participant expected holding down the mouse button in the scroll bar trough to continue moving, however the prototype did not behave in this way (an oversight of the developer). Both participants used the scroll bar thumb for the other tasks. Two participants also suggested complementing the design with 'Go to Start' and 'Go to End' buttons, as in the previous design.

Participants sometimes overshot their target, but never by much and they were able to quickly correct this. The Scroll Bar design was described by participants as "more efficient", "responsive", "faster", and participants reported feeling in control. Two participants reported that they did not like performing a lot of mouse movement however both also stated that they felt in more control using the Scroll Bar.

All participants reported being satisfied using this design, describing it as "responsive", "easier to use", and one saying that they "didn't need to think" whilst using it. One participant suggested that the design may not work so well for long sequences.

9.3.4 Connected View Design

The behaviour of the Connected View design is described in Section 7.2.3 and its appearance can be seen in Figure 53.

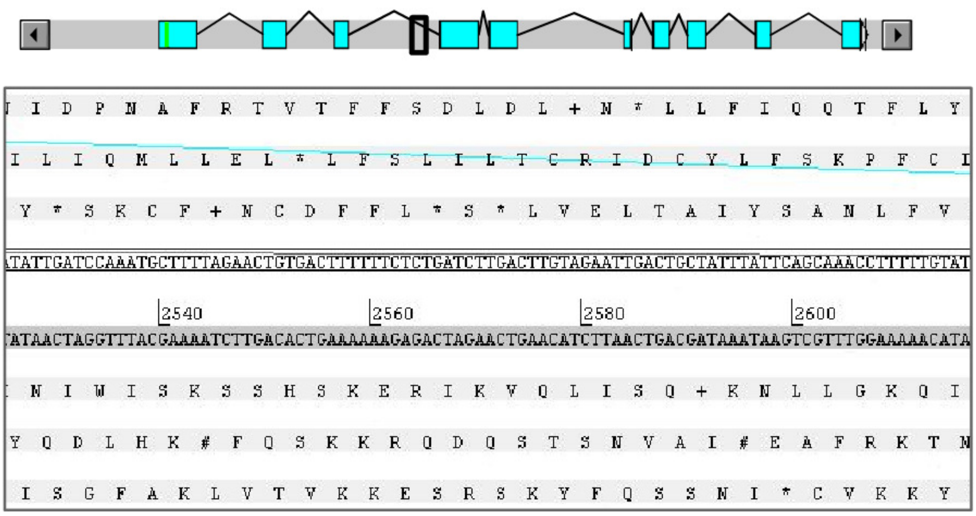


Figure 53 – Connected view as presented in the usability study

There was no common term for this design, however all participants recognised what the design represented. Terms suggested for the design were: “consensus bar”, “location finder”, “motif bar”, “summary scroll bar” and “annotated scroll bar”. Three participants suggested terms for the moveable window component, two suggested “viewfinder” and another suggested “window”.

Participants interacted with the Connected View in different ways. The use of the design was similar to that for the Scroll Bar, with dragging the window being a common behaviour. For the ‘Find Feature’ task, five participants dragged the window along the overview and two participants clicked on the overview to go to that location. For the ‘Go to a Position’ task, the two participants who had clicked on the overview during the ‘Find Feature’ task instead dragged the window or used the buttons to complete that task. All other participants dragged the window. This change in interaction style is likely because the overview did not provide exact position information that would facilitate clicking to go to a location. For the ‘Identify Features’ task, the design provided enough information in itself to complete the task and only one participant interacted with the design for that task. All participants seemed to develop a way of using the Connected View that was effective for them, with none of the participants overshooting their target.

All participants commented on the benefits of showing extra information, “it’s good [because] you can see what you’re coming up to, or go straight to where you want to go”. Several participants suggested showing different features, according to their specific needs. Another suggestion was that “location indicators” be shown on the overview, information that would have assisted in the ‘Go to a Position’ task. One participant also suggested the addition of ‘Go to Start’ and ‘Go to End’ buttons to this design.

All participants reported being satisfied with the design however the success of this design may be dependent upon the features inherent to a particular sequence. These sequences were chosen to be visually interesting – containing a selection of features – which will not always be the case. One participant stated that they expected the design would be good for long sequences also.

The Connected View design exists in other applications, as seen in the application review in Chapter Five, however this was not provided in a dynamically responsive way. Of the six applications providing a Connected View or Circular Map, none of them provided a smooth transition during the panning operation. Rather, the Detail View would ‘jump’ to the new location. The participant response to this design was positive.

9.3.5 Participant Preferences

At the end of Section Two, participants were asked to select their preferred design, for each task and for overall satisfaction. The results are shown in Table 20.

Table 20 – Number of participant's preferring each design for each task and overall

	Panning Buttons	Scroll Bar	Connected View
Find Feature	0	1	6
Go to a Position	0	1*	7*
Identify Features	1	0	6
Overall	1	0	6

* One participant could not choose between the Scroll Bar and Connected View designs for 'Go to a Position'

Most participants selected the Connected View as their preferred design for each category. One participant could not choose between the Scroll Bar and Connected View for the 'Go to a Position' task. Another participant selected the Scroll Bar for 'Find Feature' and Panning Buttons for 'Identify Features' and as their overall preference.

The divided preference for Scroll Bar for 'Go to a Position' may be explained by the fact there was no advantage to the Connected View for this task. The participant who selected the Panning Buttons design for 'Identify Features' and overall may be explained by their comment that they "don't like [...] a lot of mouse movement".

9.3.6 Other Discussion

The different tasks were completed with differing strategies. The 'Find Feature' task was typically completed by moving to the start of the sequence and then panning through the sequence until they found their target. The 'Go to a Position' task was completed by panning through the sequence until they reached the position. The 'Identify Features' task asked participants to identify features between two points. This task was typically completed by either panning backwards to the start point, or by panning to the start point and then panning forward to the end point.

Other ideas suggested by participants were keyboard interactions, mouse scroll-wheel interaction, and a "Jump To" dialogue box. One participant in particular disliked using the mouse and it is reasonable to expect keyboard interactions and shortcuts to be provided also. These were not investigated as part of this research. Likewise, computer mice with scroll-wheels are common however not ubiquitous, so it would be useful to consider how the mouse scroll wheel could be used for interaction. A dialogue box was suggested so that if the user knew the location or name of a feature they wanted to navigate to, they could go directly there.

9.3.7 Conclusion

The Connected View design was the most efficient, most preferred and most effective design of those examined for panning. Even the participant who had difficulties with other designs was able to effectively use the Connected View.

9.4 Section Three: Symbol and Orientation Meanings

This section investigates the meanings to participants of symbols and their orientation in three stages. Stage One investigated which symbols meant ‘Show more detail’, Stage Two investigated which direction (‘Up’ or ‘Down’) meant ‘Show more detail’. Stage Three combined the findings of Stages One and Two to find overall the most preferred arrangement of buttons for ‘Show more detail’.

In Stage One, participants were presented with two symbols placed on two buttons arranged horizontally and asked to click on the button that most means ‘Show more detail’ to them. The results are presented in Table 21.

Table 21 – The symbol sets as presented to participants with the most preferred symbol shaded

Left	Right	Percentage Agreement
−	+	100%
▲	▼	71%
out	in	86%
•	●	86%
1	100	57%

Participants responded well to this section. All were able to look at the symbol sets and choose a representation. Some sets required more thought than others, and these were also the sets where opinions differed.

In Stage Two, participants were presented with two unlabelled buttons arranged one above the other and asked to click on the button that most means ‘Show more detail’ to them. Only momentary thought was required for which direction represented ‘Show more detail’.

Four participants selected the 'Lower' button and three participants selected the 'Upper' button, which was inconclusive.

In Stage Three, the symbol sets were presented all at once, arranged according to each participant's preferences selected in the first two stages. For example, if a participant indicated in Stage One that '+' and in Stage Two that 'Upper' meant 'Show more detail', then the '+' symbol was presented above the '–' symbol in Stage Three. Participants were then asked to rank the symbol sets from 'best represents' to 'least represents' selecting increased level of detail. A scale from one-to-five was used to represent this, with one being 'best'. The results are presented in Table 22.

Participants rapidly decided upon their preferred representations in this last stage. The clear preference was the '+' and '–' symbol set.

Table 22 – The symbol sets as ranked by participants

Symbol Set	**Average Ranking**
+ –	1.3
in out	2.4
▼ ▲	3.0
100 1	4.0
● •	4.2

Two participants suggested a further representation, consisting of a magnifying glass with a '+' or '–' in the lens, but another participant said the '+' or '–' representation itself was "pretty standard".

9.5 Section Four: Usability Test with Designs for Zooming

Prototypes of designs for zooming were presented to participants and they were asked to perform the same tasks as in Section Two for each of these designs. These tasks are described in Section 9.3. The Scroll Bar was also presented with each of the zooming designs to allow panning. Participants were free to use, or not use, the panning and any of the zooming controls available to complete the tasks.

As in Section Two, participants quickly established what they were to do and began using the controls. A lot of feedback was gained during this section about each of the designs being proposed. The results for these designs will be examined together and then each design discussed individually.

9.5.1 Performance Measures

Each task in Section Four was completed by all participants with each of the designs. However, seven of the participants' answers to tasks were outside the expected range. Each of these errors is discussed with the design the error occurred with.

The analysis for Section Four was complicated by several factors:

- The starting sequence location for each task was not standardised by the application, most prominent for the first task,
- Limitations in the prototypes which hindered participants' performance, and
- Data that was missing because it was not recorded by the application.

Despite these issues it is instructive to look at some of the basic performance measures.

Participants had the freedom to use zooming or not in completing the tasks. A comparison of the actual use of zooming for each task is given in Table 23. It can be clearly seen that zooming was used least often for the 'Go to a Position' task. The typical user behaviour for tasks will be discussed in Section 9.5.6.

Table 23 – Percentage of participants that used zooming for each design and each task

	Find Feature	**Go to a Position**	**Identify Features**	**All Tasks**
Zoom Buttons	71%	43%	57%	57%
Zoom Slider	86%	0%	86%	57%
On-View Slider	100%	43%	86%	76%
All Designs	86%	29%	76%	

The average time to complete each task is shown in Table 24, including the average time to complete comparable tasks using the Scroll Bar design in Section Two. 'Participant A' was again excluded from these results as they took significantly more time to complete all tasks than other participants. It can be seen that for the first two tasks it took significantly longer to complete those tasks using zooming controls than with the panning control in Section Two. However the last task, 'Identify Features', was completed faster using the Zoom Buttons and Zoom Slider.

Table 24 – Average time (seconds) to complete each task, excluding Participant A

	Find Feature	**Go to a Position**	**Identify Features**
Zoom Buttons	18.0	11.1	18.9
Zoom Slider	25.7	5.0	10.0
On-View Slider	34.3	31.7	23.1
Scroll bar (from Section 2)	8.5	4.9	22.9

As participants had the freedom to use zooming or not, Table 25 presents the average time to complete each task for participants who did not use zooming during the task. Table 26 presents the average time to complete each task for participants who used zooming during the task. Interestingly, even when participants used panning only the On-view Slider seems to have had a negative impact upon performance, perhaps because it partially obscured the view.

Comparing participants who used zooming versus those who did not, there was no apparent benefit from using zooming during the tasks, which is thought to be attributable to the issues highlighted above. It took longer to complete all tasks with zooming, with the exception of 'Identify Features' with the Zoom Slider which was completed significantly faster.

Table 25 – Average time (seconds) to complete each task, for participants who panned only, excluding Participant A

	Find Feature	Go to a Position	Identify Features
Zoom Buttons	6.3	6.3	18.3
Zoom Slider	8.5	5.0	22.2
On-View Slider	N/A†	10.7	22.6
Scroll bar (from Section 2)	8.5	4.9	22.9

Table 26 – Average time (seconds) to complete each task, for participants who used zooming during the task, excluding Participant A

	Find Feature	Go to a Position	Identify Features
Zoom Buttons	23.8	15.8	19.6
Zoom Slider	29.2	N/A†	7.5
On-View Slider	34.3	52.6	23.2
Scroll bar (from Section 2)	8.5	4.9	22.9

† 'N/A' displayed in table cells where there were no participants in a category.

9.5.2 Zoom Buttons Design

The behaviour of the Zooming Buttons design is described in Section 7.3.1 and its appearance can be seen in Figure 54.

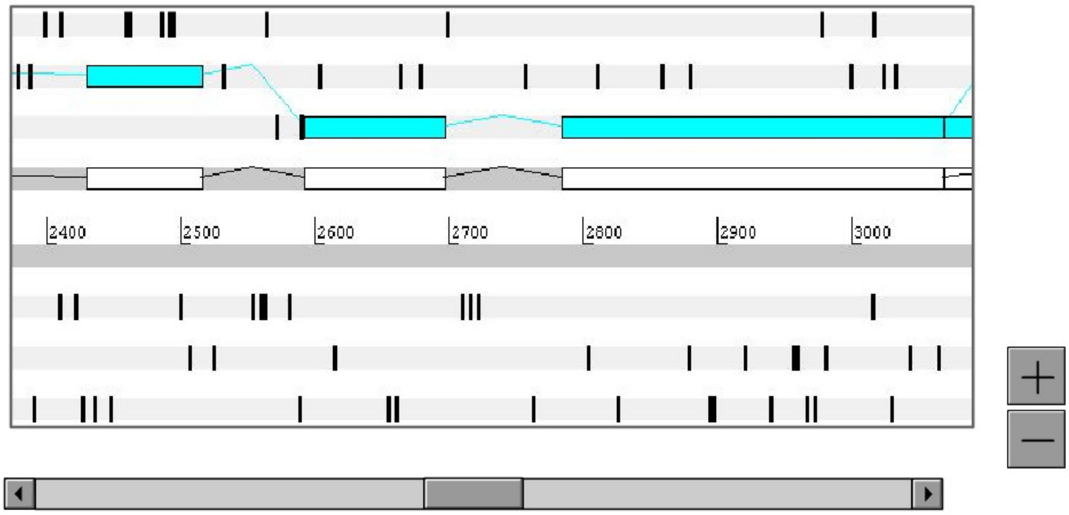


Figure 54 – Zooming buttons as presented in the usability study

Only one participant gave a name for this design, which they described as “zooming buttons”. The functionality was described by one participant as “basic” and “very standard”.

Two participants did not use zooming for navigation at all. When later asked, they said it was because they never had the [ability] in other software and so did not think to use it. One participant who did zoom stated that their “first reaction was just to scroll” as they were accustomed to just panning and that they may not have used zooming “just because of familiarity”.

One participant used the design to zoom out to view the features and then clicked on that feature, expecting to zoom in on that feature. An improvement one participant suggested was providing “different zooming ‘rates’”, presumably to offer more user control.

Two participants provided incorrect answers for the ‘Find Feature’ task. One participant’s answer was ten bases fewer than expected. This may be because the visualisation marked the sequence in intervals of twenty and that participant may have forgotten that. This was considered a trivial error. In the second incorrect answer the participant located the second feature and not the first, which was considered a serious error. The cause of this error was not evident.

9.5.3 Zoom Slider Design

The behaviour of the Zoom Slider design is described in Section 7.3.2 and its appearance can be seen in Figure 55.

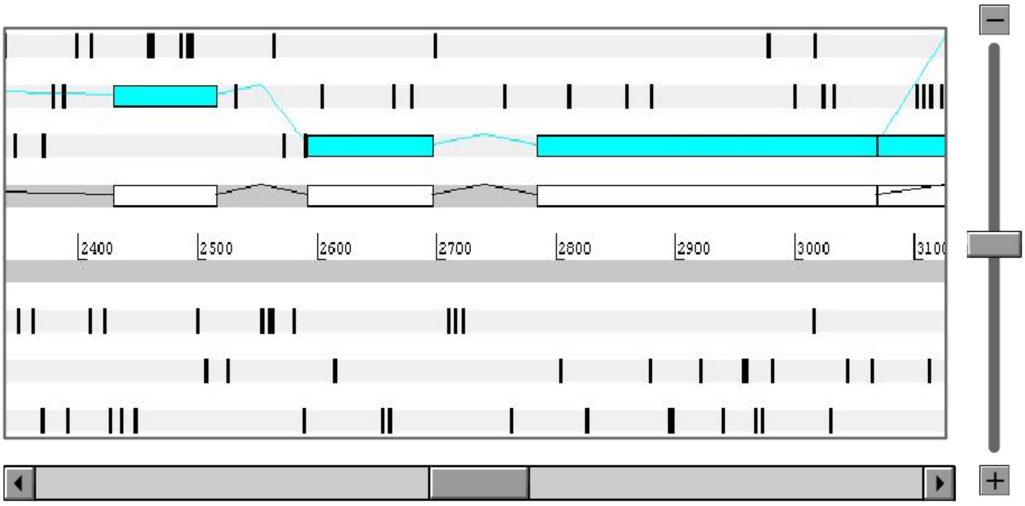


Figure 55 – Zoom Slider as presented in the usability study

Two participants gave this design the name “zoom slider” and another two gave it the name “zoom bar”. It was described as “simple” and “recognisable”. Three participants stated that they felt more in control with this design and another said it was the “easiest design with no practice”. The design was also liked for the fact that the range of levels of magnification is visible, and the current position within them.

One participant particularly disliked the orientation of the Zoom Slider but otherwise all participants seemed satisfied using it. An improvement one participant suggested was annotating the Zoom Slider to show at which levels the different features became visible, perhaps inspired by the Connected View design for panning. One participant stated that the Zoom Slider worked well in combination with the Scroll Bar.

Two participants provided incorrect answers to tasks using this design. The first error was for the ‘Go to a Position’ task, where the participant gave the answer for the location ‘3040’ instead of ‘3000’. This may have been due to the distortions of the text due to the animation technique and was considered a trivial error. The second error, by another participant, was for the ‘Identify Features’ task. The answer for “How many boxes are between the second exon and location 3000?” given was ‘none’, however an answer in the range of one-to-three was expected (depending upon interpretation). From reviewing the video, it appears that they mistook the third feature of the sequence for the second, and therefore answered for the wrong range. This error was considered serious.

9.5.4 On-view Slider Design

The behaviour of the On-view Slider design is described in Section 7.3.3 and appearance can be seen in Figure 56.

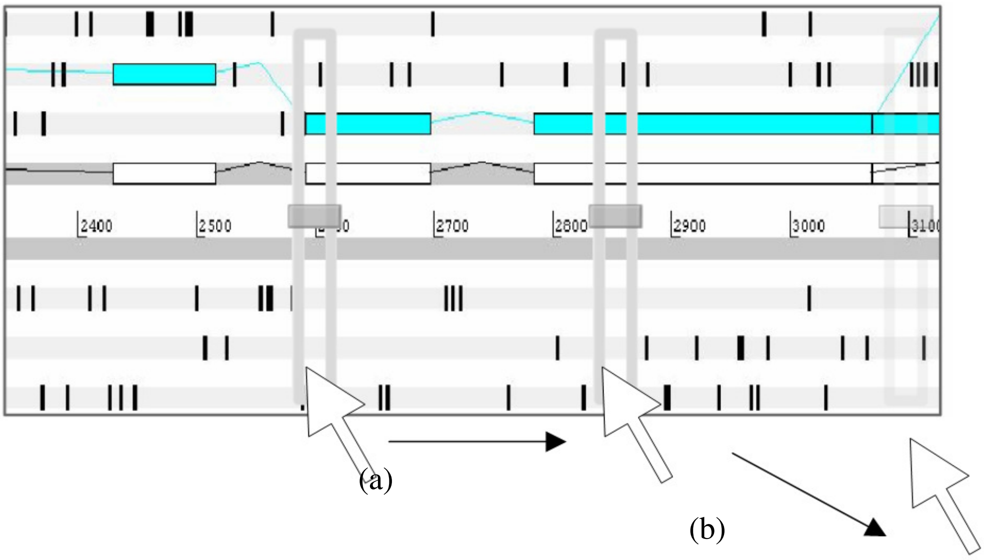


Figure 56 – On-view Slider as presented in the usability study. The behaviour is shown as (a) the mouse cursor moves in the direction shown over the view, and (b) the mouse cursor moves in the direction shown outside the view, and the On-view Slider fades. Note: in this re-creation the mouse cursor is exaggerated

There was no consensus on a term for this design, being described by three individual participants as "floating zoom tool", "dynamic slider" and "zooming dynamic scroll bar". Some participants liked this design as they experimented with it at the beginning however after using it for the tasks, no participant was satisfied with it. It was reported as "more complicated" and "distracting" because the control was over the view. Most were frustrated by the control obstructing the sequence although the vertical lines of the control were considered useful to assist in counting bases. The design was described as "frustrating", "annoying" and "difficult to understand".

Participants liked that the design was "ever present" although two participants said it was "annoying when trying to read sequences". The fading of the design when the mouse moved from the view was considered useful.

Two participants tried clicking in the trough of the slider to move the slider, another two participants tried clicking in the trough and dragging horizontally to pan the view. The design did not perform either of those actions. As two sets of participants expected different responses, neither response could be recommended from these results. One participant stated that the '+' and '–' buttons in the first two designs made their purpose self-evident but that was lacking in this design.

All participants recognised the 'side-panning' behaviour of the design when the mouse was to one side of the view, although only three participants used that during the tasks. Accidental panning proved to be annoying.

Three participants provided incorrect answers for the tasks. The first error was a data entry error (where they typed the first figure twice), and so was considered trivial. The second was for the 'Go to a Position' task, where the participant gave the answer for the location '4080' instead of '4000'. This may have been due to the distortion of the text and was considered a trivial error. The third error was for the 'Find Feature' task and the participant found the fourth feature instead of the third, however the video was not available to examine how that happened. This error was considered serious.

There was an error in the implementation of this design that seriously affected usage. This error was that panning performed by the 'side-panning' behaviour of the On-view Slider was not communicated to the Scroll Bar. The effect of this was that if the participant 'side-panned' and then used the Scroll Bar, the view would jump to the location indicated by the Scroll Bar. This seriously affected one user but was not experienced by other participants.

Overall, it was considered that this design would take some time to learn and understand.

9.5.5 Participant Preferences

At the end of Section Four, participants were asked to select their preferred design, for each task and for overall satisfaction. The results are shown in Table 27.

Table 27 – Number of participant's preferring each design for each task and overall

	Zoom Buttons	**Zoom Slider**	**On-view Slider**
Find Feature	1	4	2
Go to a Position	1	4	2
Identify Features	2	3	2
Overall	1	6	0

One participant described the Zoom Slider as "more straightforward" than the On-view Slider and also considered the Zoom Slider "better than" the Zoom Buttons. This seems to reflect the common opinion.

The participant who expressed preference for Zoom Buttons did so for all tasks and noted that, if the Zoom Slider was oriented horizontally, theirs would have been a tied preference. Another participant stated a preference for the On-view Slider for each task, but stated overall preference for the Zoom Slider.

9.5.6 Other Discussion

There were some shortfalls in the implementation of the prototypes for zooming. Participants expected the zoom to be centred on the position in the centre of the view but the prototype did not do this accurately. Also, the appearance of smooth zooming was achieved by using a disjoint set of levels and filling between those levels by compressing and expanding the images. Participants did not necessarily cease zooming at a level where the image was normal size, meaning they were often looking at a distorted image that was not clear. This may have lead to several of the errors observed.

As participants were not familiar with the concept of zooming for sequence navigation, there was some discussion on how they would use it in usual practice. One participant said that being able to zoom made navigating long sequences easier.

With zooming, the typical behaviour of participants for tasks was different again. A comparison of the use of zooming for each task is displayed in Table 23, in which it can be seen that zooming was used least often for the ‘Go to a Position’ task. For the ‘Find Feature’ task, the typical behaviour of participants was to zoom out, pan to find the feature and then zoom in on that feature (Figure 57). The ‘Go to a Position’ task was usually completed by simply panning through the sequence until they reached the position. For the ‘Identify features’ task participants typically simply zoomed out.

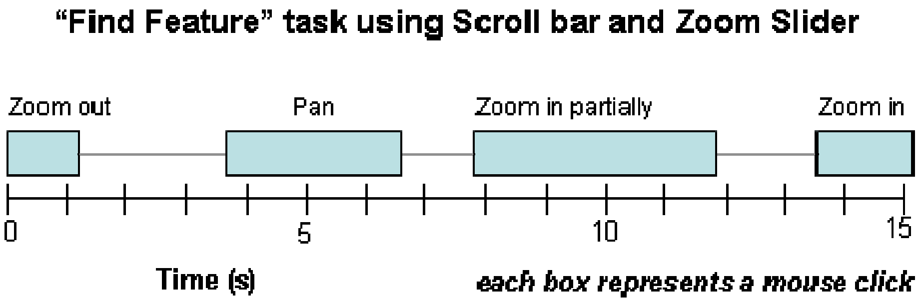


Figure 57 – Showing typical behaviour for the ‘Find Feature’ task

The participant whose behaviour is exhibited in Figure 57 started the task by zooming out to near overview-level. They then panned the sequence to be centred on the feature, and zoomed in to read the base numbers. They zoomed in using two operations, the first partially zooming in, the second time zooming to detail-level. This would appear to be an efficient way to navigate the sequence for this navigational activity. However, as the view conditions were not standardised for participants at the start of this task, accurate data is not available.

An error considered ‘serious’ was witnessed with each of these designs for zooming, each by different participants. Given that the data available does not make it clear whether the serious errors were caused by participant error or by the design, they cannot be investigated further.

Other ideas suggested by participants were:

- Speed-dependent automatic zooming, where the view is generally displayed at detail-level, however as the view is panned it also zooms out according to the speed.
- Clicking on the view to zoom (as a direct manipulation), but that participant also noted the difficulties of how to zoom out and to differentiate that from other clicking operations.
- Interactions with features, so that when a user clicks on a feature in the view that feature is zoomed on so that it fills the view.
- Incorporating zoom with the panning control, with the mouse direction indicating action (perhaps the same as dynamic zooming).

9.5.7 Conclusion

The Zoom Slider design was easily understood and participants reported feeling “more in control”. The time to perform the ‘Identify Features’ task using the Zoom Slider design was significantly less than with other zooming designs and panning-alone. It was also the most preferred design as reported by participants and considered to work well with the Scroll Bar design.

9.6 Section Five: Brief Investigation of Other Interfaces

Prototypes of two other designs for panning and zooming were presented to participants although there were no set tasks for participants to perform. It was considered that these designs did not warrant a full usability evaluation but may benefit users in special cases. As there were no set tasks, there are no performance measures and an entirely descriptive approach is taken.

9.6.1 Dynamic Zoom

The behaviour of this design is described in Section 7.4.1. There was no visible control. Four participants said that they liked the Dynamic Zoom design, two did not like it, and one “[didn’t] mind it”. Those who liked it said that it would be “intuitive once you get [used to] it”, that it was simple and allowed screen-space to be used for other things. Those who disliked the design said it would be “too much work [and] you need to be steady to be good at it”, and that they didn’t intuitively know how to use it.

Participants found the design very confusing at first; a common question was “How do you know how to use [this]?” The design was also reported to be very mouse-intensive, particularly for panning. One participant suggested providing a Scroll Bar for panning-only and Dynamic Zoom for zooming-only. Another problem encountered was that small accidental movements often completely changed the view.

From witnessing the use of this interaction method, it would seem very difficult to find the correct balance that prevents accidental movements without making intentional movements tiresome. It would appear that Dynamic Zoom does not meet the attributes of usability.

9.6.2 Magnifying Glass Tool

The behaviour of this design is described in Section 7.4.2, and buttons were used to select a mouse mode (Figure 58) and the mouse cursor indicated which mode was in use (Figure 59).

(a) (b) (c)

Figure 58 – Buttons provided to select mode: (a) zoom in, (b) zoom out, (c) pan

(a) (b) (c)

Figure 59 – Mouse cursor symbols used to portray mode: (a) zoom in, (b) zoom out, (c) pan

Participants found it was confusing to know how to zoom out again and reported that it was slower than other techniques. One participant reported a preference for being able to see contextual information such as size of the space and position within that. Several participants expressed expectations different from the actual design, for example expecting to be able to click and drag the view to pan it.

Overall, participants were neutral about this design and it would not appear to meet the attributes of usability.

9.7 Participant Feedback

During the study, participants gave feedback on the visualisation, tasks and sequences involved in the study. This feedback was in the form of comments during and following the tasks.

9.7.1 Visualisation

Participants were asked at the end of Section Two for their opinion on the visualisation and all reported being satisfied. Zooming interactivity was introduced in Section Four of the study and participants were again generally satisfied with the implementation of this, although some commented that the transition between magnification levels was quite sudden. Also, it was noticed that the text at some levels of magnification became fuzzy. Only one participant described this as a problem however it appears to have contributed to at least two of the incorrect answers that participants gave to the tasks. Two participants stated that the visualisation needs to remain centred on the same point during zooming.

9.7.2 Tasks

Participants were asked at the end of the trial for their opinion on the tasks performed within the trial. Most participants considered that the tasks were "realistic", one participant said that "anyone with at least basic sequence work will have done them", another that the tasks were "everyday stuff". Two participants suggested more complex tasks that could have been trialled such as selecting a region of the sequence, creating annotations and aligning multiple sequences.

Two of the tasks were frequently misinterpreted by participants. The question "What is the location of the first base…" was frequently read as "What is the first base…" Confusion also occurred with "How many boxes are there…" at which point participants asked, "What are the boxes?" They had already accepted them as representing features or "exons", so asking "How many features" would have been clearer. Participants quickly realised their error with these tasks when the study application did not progress to the next task, and they re-read the question. This will not have affected any of the performance timing measures displayed above. These issues would have been identified in pre-testing by users who knew the terminology of the bioinformatics domain, with the disadvantage that that person could not have been used for a usability trial.

9.7.3 Sequences

Participants were asked the nature of the sequences that they typically worked with and to consider whether the visualisation would be suitable for them. They reported working with sequences of varying lengths. Five participants worked with sequences of the same scale of those tested (several thousand base-letters long). One participant worked with sequences much shorter (several hundred base-letters long), and one participant worked with sequences much longer (several hundred thousand to several million base-letters long). All participants indicated that the visualisation and interaction designs should be suitable for the sequences they work with.

Chapter 10: Conclusions

Visualisation and interaction with data is vital to bioinformatics and the aim of this research was to investigate aspects of it. The objective was to develop and evaluate ways of displaying and navigating genetic sequences on computer for use by bioinformatics researchers.

This research was carried out using a series of steps:

1. A questionnaire was used to investigate the target users, their work environment and the tasks that they perform;
2. Current bioinformatics applications were examined to identify existing designs for displaying the information and facilitating navigation;
3. Heuristic evaluation was used to critically analyse the designs identified in the applications and a smaller set of designs was selected for further evaluation; and finally
4. Prototypes of the selected designs were developed and presented to users in usability testing.

This chapter will summarise the results and issues that have been encountered during the process of this research and make some recommendations.

10.1 Summary of Results

The purpose of this research was to examine usability, the attributes of which were described in Section 2.4.1 in terms of understandability, efficiency, effectiveness and user satisfaction. This research was undertaken to find out what users understand and what they find efficient, effective and satisfying.

The questionnaire assisted in the development of an understanding of the target user, the environment in which they work, the tools they use and the tasks that they perform. Also, it was found that respondents often had to work with sequences on paper, even though none of them preferred to do so. Examining existing applications in the bioinformatics domain was instructive and highlighted the need for the application of user-centred design activities in development. There was no predominant method for panning or zooming, and it is interesting that the use of smooth transitions is uncommon in existing applications despite the evidence showing the advantages they provide in assisting spatial navigation. Heuristic evaluation was useful in deciding which designs to further investigate. It suggested fewer issues for the scroll bar and slider designs which proved to be true in the usability study. Also, participant's comments during the usability study were consistent with the issues identified.

The process of developing the prototypes was quite time-consuming and the focus of these had been tidy appearance, expected behaviours and development of generic, reusable components. It is considered in hindsight that "quick and dirty" prototypes would have been sufficient however it is also important that prototypes accurately replicate the intended behaviours of the design.

The usability study was crucial in understanding the actual user experience. Observations and comments from participants reinforce the need to provide contextual information and other 'landmarks' where possible. Of the designs examined for panning, the Connected View design was the most efficient, most preferred and most effective design. Even the one participant who had difficulties with other designs was able to effectively use the Connected View. Of the designs for zooming, the Zoom Slider design was easily understood and participants reported feeling "more in control". The time to perform the 'Identify Features' task with this design was significantly less than with other zooming designs or panning alone. It was also the most preferred design for zooming and was considered to work well with the Scroll Bar design for panning.

The usability study also gave useful information on the ways that users approach tasks. Some tasks suited the use of zooming, particularly those involving browsing, and others were better completed using panning alone. For the 'Find Feature' task, the typical behaviour of participants was to move to the start of the sequence and then pan through the sequence until they found their target. When zooming was available, users would zoom out, pan to find the feature then zoom in on that feature. The 'Go to a Position' task was usually completed by simply panning through the sequence until the position was reached. For the 'Identify features' task participants simply zoomed out. If zooming was not available, they would pan through the range.

The use of zooming may take time for users to adjust to, although the growth of applications such as Google Maps will benefit users' understanding.

10.2 Limitations

Limitations with this research that have been identified were that:

- It would have been helpful to have had more respondents to the questionnaire, enabling more generalisable results,
- It would have been useful to interview a small number of target users to gain a more detailed understanding of their software needs as this was not well answered in the questionnaire,
- Having more than one evaluator for the heuristic evaluation may have produced more comprehensive findings,
- All designs and all tasks were presented to participants in the same order each time meaning that participants may have been affected by the previous tasks and designs,
- The starting conditions for each task were not standardised, so participants did not necessarily start at the same position or level of magnification. This would have been most prominent for the first task as later tasks would have been standardised by the end conditions of the previous task,
- The visualisation did not accurately centre the zooming operation on the centre of the view and the On-view Slider prototype did not interact appropriately with the Scroll Bar design it was presented with, and
- Having more people to facilitate and observe the usability study would allow more comprehensive notes and findings.

Participants comments during the usability study that highlighted other limitations were that sequences are “not always so annotated” (i.e. containing so many features) and that the application presented was suited for viewing results only, rather than performing analyses.

10.3 Future Work

There is much more potential research arising from this study. As noted in the limitations, the application presented in the usability study was suited for viewing results, whereas respondents to the questionnaire listed many other tasks they also perform. These include the annotation of genetic sequences, alignment of multiple sequences and analysis of subsections of a sequence. Working with that type of data on computer would also introduce challenges as to how it is organised and displayed (for example, tracks of annotations) and how it can be manipulated (for example, methods of filtering the data).

Smooth transitions have not been utilised in existing bioinformatics applications, and parameters for this need further investigation. For example, the use of variable speeds in transitions, the number of levels of magnification provided and the speeds of panning and zooming.

In terms of user interface design, there is potential for further examination of some of the designs explored in this study. The buttons used exhibited complex behaviours (for example, variable speeds) and how users learn the inherent parameters of such controls could be investigated. It would be beneficial to develop Connected Views further as this design showed much promise and was well liked by users.

Most important is the continued research into the actual understandings and expectations that biologists have of the applications developed for them. Without this information, applications cannot be developed that are truly usable by that audience.

10.4 Final Comments

It is considered that the overall aim, to develop and evaluate ways of displaying and navigating genetic sequences on computer for use by bioinformatics researchers, has been achieved. Each step of the process in implementation contributed towards this overall objective.

The overall recommendation from this research would be the use of user-centred design in development of applications for biologists. A general recommendation would be the development of Connected Views for use in bioinformatics applications. In the absence of Connected Views, Scroll Bars also perform well. For the purpose of selecting level of magnification, the Zoom Slider design provides an easily understood metaphor for doing this. The use of smooth transitions is considered to assist users in navigation of the data and should be considered in future designs.

“Software developers need to go beyond user-friendliness to a biologist-centric approach for building tools for researchers” (Kumar et al., 2008). This research can be seen as a useful contribution towards this goal.

[This page is intentionally blank]

References

Abratt, D., Mallinson, W., & Bekker, A. (2003). *Usability Testing: Recipe for Success*. Paper presented at the STARWEST: International Conference on Software Testing Analysis and Review.

Adobe. (2004). Adobe Acrobat Reader

Apple Computer, (2004). *Apple Human Interface Guidelines*.

Applied Biosystems. (2004). SeqScape 2.5

Bederson, B. B., & Boltman, A. (1998). Does Animation Help Users Build Mental Maps of Spatial Information? *HCIL Technical Report, 98-11*

Benyon, D., & Höök, K. (1997). Navigation in information spaces: supporting the individual. *Proceedings of INTERACT'97, London: Chapman and Hall*, 39-46.

Bevan, N. (1999). Quality in use for all. In C. Stephanidis (Ed.), *User Interfaces for All*: Lawrence Erlbaum.

Bevan, N., & Macleod, M. (1994). Usability measurement in context. *Behaviour & Information Technology, 13*, 132-145.

Bisson, G., & Garreau, A. (1995). *APIC -- a generic interface for sequencing projects*. Paper presented at the Proceedings of the International Conference on Intelligent Systems for Molecular Biology

Bonfield, J. K., Smith, K. F., & Staden, R. (1995). A new DNA sequence assembly program. *Nucleic Acids Research, 23*(24), 4992-4999.

Card, S. K., Mackinlay, J. D., & Shneiderman, B. (Eds.). (1999). *Readings in information visualization : using vision to think*. San Francisco, California: Morgan Kaufmann Publishers.

Chen, Y. B., Chattopadhyay, A., Bergen, P., Gadd, C., & Tannery, N. (2007). The Online Bioinformatics Resources Collection at the University of Pittsburgh Health Sciences Library System--a one-stop gateway to online bioinformatics databases and software tools. *Nucleic Acids Research, 35*(Database issue), D780.

Cheng, L.-T., & Gruen, D. (2003). *A Mobile User Interface for Threading, Marking, and Previewing Email, Technical Report 03-08*. Cambridge, MA 02142: IBM Watson Research Center.

Cockton, G., & Woolrych, A. (2002). Sale Must End: Should Discount Methods be Cleared off HCI's Shelves? *Interactions, 9*(5), 13-17.

Cohen, J. (2004). Bioinformatics—an introduction for computer scientists. *ACM Computing Surveys (CSUR), 36*(2), 122-158.

Cohen, J. (2005). Computers Science and Bioinformatics. *Communications of the ACM, 48*(3), 72-78.

Corel Corporation. (2003). Corel PHOTO-PAINT 12.0. Canada

Crapo, A. W., Waisel, L. B., Wallace, W. A., & Willemain, T. R. (2000). *Visualization and the process of modeling: a cognitive-theoretic view*. Paper presented at the Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining.

Dahlbäck, N. (1998). On spaces and navigation in and out of the computer. *Exploring Navigation; Towards a Framework for Design and Evaulation of Navigation in Electronic Spaces. Technical Report T, 98*, 15-29.

Dahlberg, A., & Götesson, A. (2004). *Bridging the Gap.* Göteborg University and Chalmers University of Technology, Göteborg, Sweden.

Dillon, A., Richardson, J., & McKnight, C. (1990). Navigation in Hypertext: a critical review of the concept. In D.Diaper, D.Gilmore, G.Cockton & B.Shackel (Eds.), *Human-Computer Interaction-INTERACT'90* (pp. 587-592). North Holland: Amsterdam.

Doerry, E., Douglas, S., Kirkpatrick, T., & Westerfield, M. (1997a). Task-centered Navigation in a Web-accessible Dataspace. *Proceedings from the AACE/WebNet97 Conference,*

Doerry, E., Douglas, S. A., Kirkpatrick, A. E., & Westerfield, M. (1997b). Participatory Design for Widely-Distributed Scientific Communities. *Proceedings of the Third Conference on Human Factors and the Web, Denver, CO,*

ECan. (2005). *ECan Online GIS Viewer*. Retrieved 21/09/2005, 2005, from http://www.ecan.govt.nz/ecangis/ecanpro/viewer.htm

ESRI. (2005). Arc Internet Map Server

Federoff, M. A. (2002). *Heuristics And Usability Guidelines For The Creation And Evaluation Of Fun In Video Games.* Indiana University.

Forment, J., Gilabert, F., Robles, A., Conejero, V., Nuez, F., & Blanca, J. M. (2008). EST2uni: an open, parallel tool for automated EST analysis and database creation, with a data mining web interface and microarray expression data integration. *BMC Bioinformatics, 9*(1), 5.

Fox, J. A., McMillan, S., & Ouellette, B. F. (2007). Conducting Research on the Web: 2007 Update for the Bioinformatics Links Directory. *Nucleic Acids Research, 35*(Web Server issue), W3.

Fuchs, R. (2000). Analyse this... or: intelligent help for the rest of us. *Bioinformatics, 16*(6), 491-493.

Gene Codes Corporation. (2003). Sequencher 4.2: Ann Arbor, Michigan

Glusman, G., & Lancet, D. (2000). GESTALT: a workbench for automatic integration and visualization of large-scale genomic sequence analyses. *Bioinformatics, 16*(5), 482-483. Applications Note

Google. (2005). *Google Maps*. Retrieved 21/09/2005, 2005, from http://maps.google.com

Guiard, Y., Beaudouin-Lafon, M., Bastin, J., Pasveer, D., & Zhai, S. (2004). View size and pointing difficulty in multi-scale navigation. *Proceedings of the working conference on Advanced visual interfaces*, 117-124.

Gutwin, C. (2005, 19/01/2005). In N. Z. Presentation at University of Canterbury (Ed.). Christchurch, New Zealand. Presentation

Harris, N. L. (1997). Genotator: A Workbench for Sequence Annotation. *Genome Research, 7*(7), 754-762.

Hearst, M. A., Divoli, A., Guturu, H., Ksikes, A., Nakov, P., Wooldridge, M. A., et al. (2007). BioText Search Engine: beyond abstract search. *Bioinformatics, 23*(16), 2196.

Helt, G., Lewis, S., Loraine, A., & Rubin, G. (1998). BioViews: Java-based tools for genomic data visualization. *Genome Research, 8*(3), 291-305.

Hilbert, D. M., & Redmiles, D. F. (2000). Extracting usability information from user interface events. *ACM Computing Surveys, 32*, 384-421.

Hornbæk, K., Bederson, B. B., & Plaisant, C. (2002). Navigation Patterns and Usability of Zoomable User Interfaces With and Without an Overview. *ACM Transactions on Computer-Human Interaction, 9*(4), 362-389.

Hu, Z., Frith, M., Niu, T., & Weng, Z. (2003). SeqVISTA: a graphical tool for sequence feature visualization and comparison. *BMC Bioinformatics, 4*(1)

Hubbard, T., Barker, D., Birney, E., Cameron, G., Chen, Y., Clark, L., et al. (2002). The Ensembl genome database project. *Nucleic Acids Research, 30*(1), 38-41.

Ivory, M. Y., & Hearst, M. A. (2001). The State of the Art in Automating Usability Evaluation of User Interfaces. *ACM Computing Surveys, 33*(4), 470-516.

Jamieson, S. (2004). Likert scales: how to (ab) use them. *Medical Education, 38*(12), 1217-1218.

Javahery, H., Seffah, A., & Radhakrishnan, T. (2004). Beyond power: making bioinformatics tools user-centered. *Communications of the ACM 47*(11), 58-63.

Jeffries, R., Miller, J. R., Wharton, C., & Uyeda, K. (1991). *User interface evaluation in the real world: a comparison of four techniques.* Paper presented at the Human factors in computing systems: Reaching through technology.

Jog, N., & Shneiderman, B. (1994, March 27-29,1995). *Starfield Information Visualization with Interactive Smooth Zooming.* Paper presented at the IFIP 2.6 Visual Databases Systems Proceedings, Lausanne, Switzerland. Retrieved from http://www.cs.umd.edu/local-cgi-bin/hcil/rr.pl?number=94-06

John, B. E., & Marks, S. J. (1997). Tracking the effectiveness of usability evaluation methods. *Behaviour and Information Technology, 16*(188-202)

Karabeg, A., Akkøk, M. N., & Kristensen, K. (2004). *Towards a Language for Talking about Information Visualization Aimed at Presentation on the Web*. Paper presented at the Proceedings of the Information Visualisation, Eighth International Conference.

Karat, J. (1997). Evolving the scope of user-centered design. *Communications of the ACM, 40*(7), 33-38.

Karolchik, D., Baertsch, R., Diekhans, M., Furey, T. S., Hinrichs, A., Lu, Y. T., et al. (2002). The UCSC Genome Browser Database. *Nucleic Acids Research, 2003*(31), 1.

Kelley, T., & Allender, L. (1995). Why choose? A process approach to usability testing. In K. O. Yuichiro Anzai, & Hirohiko Mori (Eds.) (Ed.), *Symbiosis of human and artifact: Human and social aspects of human-computer interaction proceedings of the sixth international conference on human computer interaction, Tokyo, Japan, 9-14 July 1995* (Vol. 2, pp. 393-398). Amsterdam: Elsevier. .

Kumar, S., Nei, M., Dudley, J., & Tamura, K. (2008). MEGA: A biologist-centric software for evolutionary analysis of DNA and protein sequences. *Briefings in Bioinformatics, 9*(4), 299.

Kuniavsky, M. (2003). *Observing the user experience : a practitioner's guide to user research*. San Francisco, CA: Morgan Kaufmann Publishers.

Lauesen, S. (2005). *User Interface Design: A Software Engineering Perspective*: Addison-Wesley.

Leung, Y. K., & Apperley, M. D. (1994). A Review and Taxonomy of Distortion-Oriented Presentation Techniques. *ACM Transactions on Computer-Human Interaction, 1*(2), 126-160.

Lewis, S. E., Searle, S. M. J., Harris, N., Gibson, M., Iyer, V., Richter, J., et al. (2002). Apollo: a sequence annotation editor. *Genome Biology, 3*(12), research0082.0081–0082.0014.

Liu, C., Bonner, T. I., Nguyen, T., Lyons, J. L., Christian, S. L., & Gershon, E. S. (2003). DNannotator: annotation software tool kit for regional genomic sequences. *Nucleic Acids Research, 31*(13), 3729-3735.

Loraine, A. E., & Helt, G. A. (2002). Visualizing the genome: techniques for presenting human genome data and annotations. *BMC Bioinformatics, 3*(19)

Luscombe, N., Greenbaum, D., & Gerstein, M. (2001). What is Bioinformatics? A Proposed Definition and Overview of the Field *Methods of Information in Science, 40*, 346-358.

Macleod, M., Bowden, R., Bevan, N., & Curson, I. (1997). The MUSiC performance measurement method. *Behaviour & Information Technology, 16*(4-5), 279-293.

MacMullen, W. J., & Denn, S. O. (2005). Information problems in molecular biology and bioinformatics. *Journal of the American Society for Information Science and Technology, 56*(5), 447-456.

Macromedia. (2004). Macromedia Flash MX. San Francisco, California

McGinnis, S., & Madden, T. L. (2004). BLAST: at the core of a powerful and diverse set of sequence analysis tools. *Nucleic Acids Research, 32*(Web Server Issue), W20.

Medigue, C., Rechenmann, F., Danchin, A., & Viari, A. (1999). Imagene: an integrated computer environment for sequence annotation and analysis. *Bioinformatics, 15*(1), 2-15.

Molidor, R., Sturn, A., Maurer, M., & Trajanoski, Z. (2003). New trends in bioinformatics: from genome sequence to personalized medicine. *Experimental Gerontology, 38*(10), 1031-1036.

Morris, M. G., & Dillon, A. (1997). How user perceptions influence software use. *IEEE Software, 14*(4), 58-65.

Myers, C. L., Chen, X., & Troyanskaya, O. G. (2005). Visualization-based discovery and analysis of genomic aberrations in microarray data. *BMC Bioinformatics 6*, 146.

Nahum, G., & Ward, P. (2001). What storytelling can do for information visualization. *Communications of the ACM, 44*(8), 31-37. doi:http://doi.acm.org/10.1145/381641.381653

National Center for Biotechnology Information. (2008). *Growth of Genbank*. Retrieved 24 April 2008, 2008, from ftp://ftp.ncbi.nih.gov/genbank/gbrel.txt

Nielsen, J. (1994). Heuristic Evaluation: How to conduct a heuristic evaluation. Retrieved from http://www.useit.com/papers/heuristic/heuristic_evaluation.html

Nielsen, J. (1997).Severity Ratings for Usability Problems http://www.useit.com/papers/heuristic/severityrating.html

Nielsen, J. (1988). Coordinating User Interfaces for Consistency. *SIGCHI Bulletin, 20*(3), 63-65. Workshop

Nielsen, J. (1990). The art of navigating through hypertext. *Communications of the ACM 33*(3), 296 - 310.

Nielsen, J. (1992). *Finding Usability Problems Through Heuristic Evaluation.* Paper presented at the Proceedings of the ACM CHI'92 Conference on Human Factors in Computer Systems.

Nielsen, J. (1993). *Usability engineering*. San Francisco, California: Morgan Kaufmann.

Nielsen, J. (1994a). *Enhancing the explanatory power of usability heuristics*. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence.

Nielsen, J. (1994b). Heuristic Evaluation. In J. Nielsen & R. Mack (Eds.), *Usability Inspection Methods* (pp. 25-62): John Wiley and Sons, Inc.

Nielsen, J., & Landauer, T. K. (1993). A mathematical model of the finding of usability problems. *Proceedings of the SIGCHI conference on Human factors in computing systems* 206-213.

Nielsen, J., & Levy, J. (1994). Measuring usability - preference vs. performance. *Communications of the ACM, 37*(4), 66-75.

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, 249-256.

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

Öörni, K. (2003). What do we know about usability evaluation? A critical view. *Conference on Users in the Electronic Information Environments, September*, 8-9.

Otter, M., & Johnson, H. (2000). Lost in hyperspace: metrics and mental models. *Interacting with Computers, 13*(1), 1-40.

Plaisant, C., Carr, D., & Shneiderman, B. (1995). Image-browser taxonomy and guidelines for designers. *IEEE Software, 12*(2), 21-32.

Plaisant, C., Milash, B., Rose, A., Widoff, S., & Shneiderman, B. (1996). *LifeLines: Visualizing Personal Histories.* Paper presented at the CHI '96.

Purves, W. K., Orians, G.H., & Heller, H.C. (1995). *Life: The science of biology, 4th ed.* Utah: W.H. Freeman and Co.

Riihiaho, S. (2000). *Experiences with usability evaluation methods.* Unpublished Licentiate's thesis, Helsinki University of Technology.

Robinson, A. J., & Flores, T. (1997). Novel Techniques for Visualising Biological Information. *Proceedings of the Fifth International Conference on Intelligent Systems in Molecular Biology, AAAI Press*, 241-249.

Roos, D. (2001). Bioinformatics--trying to swim in a sea of data. *Science, 291*(5507), 1260-1261.

Roth, A. (1999). Using the Iced T Model to Test Subjective Software Qualities. *Proceedings of the 1999 International Conference on Software Testing, Analysis and Review held in San José, California,*

Rozen, S., & Skaletsky, H. (2000). Primer3 on the WWW for general users and for biologist programmers. *Methods Mol Biol, 132*(3), 365-386.

Rudd, J., Stern, K., & Isensee, S. (1996). Low vs. high-fidelity prototyping debate. *interactions, 3*(1), 76-85.

Rutherford, K., Parkhill, J., Crook, J., Horsnell, T., Rice, P., Rajandream, M.-A., et al. (2000). Artemis: sequence visualization and annotation. *Bioinformatics, 16*(10), 944-945. Retrieved from http://bioinformatics.oupjournals.org/cgi/content/abstract/16/10/944

Rutherford, P. (2003). *A Tool for Dynamic Exploration of Primers.* Unpublished A dissertation submitted in partial fulfilment of the requirements for the Degree of Bachelor of Applied Computing with Honours, Lincoln University.

Sahlin, D. (2003). *How To Do Everything with Adobe Acrobat 6.0*: McGraw-Hill Professional.

Salgado, H., Santos-Zavaleta, A., Gama-Castro, S., Millán-Zárate, D., Díaz-Peredo, E., Sánchez-Solano, F., et al. (2001). RegulonDB (version 3.2): transcriptional regulation and operon organization in Escherichia coli K-12. *Nucleic Acids Research, 29*(1), 72-74.

Shneiderman, B. (1996). *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations.* Paper presented at the IEEE Symposium on Visual Languages, Boulder, Colorado.

Shneiderman, B. (1997). *Direct manipulation for comprehensible, predictable and controllable user interfaces*. Paper presented at the Proceedings of the 2nd international conference on Intelligent user interfaces.

Snyder, C. (2003). *Paper Prototyping: the fast and easy way to design and refine user interfaces*: Elsevier Science (USA).

Soderston, C., & Rauch, T. (1996). The case for user-centered design. *Proceedings of the STC 44th Annual Conference. Society for Technical Communication,*

Spence, R. (1999). A Framework For Navigation. *International Journal of Human-Computer Studies, 51*, 919-945.

Spence, R. (2001). *Information Visualization*. New York: ACM Press.

TechSmith. Camtasia (Version 3.0.1)

Tidwell, J. (2005). *Designing Interfaces*: O'Reilly Media, Inc.

Vernikos, G. S., Gkogkas, C. G., Promponas, V. J., & Hamodrakas, S. J. (2003). GeneViTo: Visualizing gene-product functional and structural features in genomic datasets. *BMC Bioinformatics, 4*(53)

Vincze, T., Posfai, J., & Roberts, R. J. (2003). NEBcutter: a program to cleave DNA with restriction enzymes. *Nucleic Acids Research, 31*(13), 3688-3691.

Virzi, R. A., Sokolov, J. L., & Karis, D. (1996). *Usability problem identification using both low- and high-fidelity prototypes*. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems: common ground.

Vredenburg, K., Mao, J. Y., Smith, P. W., & Carey, T. (2002). A survey of user-centered design practice. *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, 471-478.

WanAdilah, W. A., NorLaila, M. N., & Rasimah, A. (2003). Information Visualization Concepts and Issues. *7th Pacific Asia Conference on Information Systems, 10-13 July 2003, Adelaide, South Australia*, 1782-1795.

Webb, P., Alverson, J., Clark, B., Green, D., Greene, C., Kahl, C., et al. (2005, 11/01/2005). *WebVoyáge Usability Study Task Force (WVUSTF): Final Report*.

Wheeler, D. L., Barrett, T., Benson, D. A., Bryant, S. H., Canese, K., Church, D. M., et al. (2005). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research, 33*(Database Issue), D39.

Whittaker, S., Terveen, L., & Nardi, B. A. (2000). Let's stop pushing the envelope and start addressing it: a reference task agenda for HCI. *Human Computer Interaction, 15*(2), 75-106.

Wigdor, D. (2002, September 2002). *Building a Usability Prototype in Visual Basic, Flash, and Dreamweaver: A How-To Guide for CSC318*: University of Toronto, Department of Computer Science.

Woffelman, C. (2004). DNAMAN for Windows, Version 5.2.10: Lynon Biosoft, Institute of Molecular Plant Sciences, Leiden University, Netherlands

Appendices

Appendix A: Questionnaire

Appendix B: Summary of applications examined

Appendix C: Usability Study Recruitment Letter

Appendix D: Usability Study Consent Form

Appendix E: Usability Study Script

Appendix F: Usability Study Task Satisfaction Questionnaire

Appendix G: Usability Study Observer Sheet

Appendix A

Survey

You are invited to participate in a project called "A dynamic and continuously zoomable genetic sequence visualisation". We would like you to complete the following survey which is confidential to the researcher and supervisors. We want to develop an application for working with data about genetic sequences that is easy for people like yourself to use. This survey is to help us understand the requirements and expectations of people who work with genetic sequences on computers. You are under no obligation to participate, but your cooperation in this project would assist in this research. Your participation in this survey is important to us.

If you complete the survey it will be understood that you have consented to participate in the project and consent to publication of the results of the project with the understanding that anonymity of participants will be preserved.

This survey is being administered by Paul Rutherford as part of a Masters Thesis at Lincoln University. If you have any questions or comments about this survey you may write them at the end of the survey, or e-mail them to **RutherfordP@crop.cri.nz**

Please return completed surveys to John McCallum.

Contact Details

Paul Rutherford (Masters student, Lincoln University)
E-mail: RutherfordP@crop.cri.nz Phone: 325-2811 ext 8813

Clare Churcher (Supervisor, Lincoln University)
E-mail: Churcher@lincoln.ac.nz Phone: 325-2811 ext 8905

John McCallum (Associate Supervisor, Crop & Food Research)
E-mail: McCallumJ@crop.cri.nz Phone: 325 6400 ext 3466

Question 1. Please describe your job:

Question 2. How often do you use a computer?

- ○ Less than once a week
- ○ Several times a week
- ○ Every day
- ○ Several times a day, or most of the day

Question 3. What is your experience with computers? (Tick those that apply)

- □ I can use the World Wide Web (WWW) to search for information
- □ I can use a word processor to produce formatted documents
- □ I can use a spreadsheet to perform simple calculations
- □ I can produce reports with a database
- □ I can design tables and queries for a database
- □ I can perform simple programming tasks
- □ I can perform advanced programming tasks

Question 4. How often do you work with genetic sequence data?

- ○ Less than once a week
- ○ Several times a week
- ○ Every day
- ○ Several times a day, or most of the day

Question 5. How often do you use a computer for bioinformatics tasks?

- ○ Less than once a week
- ○ Several times a week
- ○ Every day
- ○ Several times a day, or most of the day

Sometimes it seems that it is easier to print out the results of a bioinformatics program to work on paper rather than continue working with it on the computer.

Question 6. How often would you choose to work on paper instead of on a computer screen for bioinformatics tasks?

- ○ Always
- ○ Most often
- ○ Sometimes
- ○ Never

Question 7. Please give your reasons for why you choose to work on paper:

Question 8. How do you store genetic sequences that you have obtained in your research? (Tick those that apply)

- □ in a database
- □ in Genbank
- □ on paper (printed)
- □ in a spreadsheet
- □ in a text file
- □ in a Word document
- □ other: ____________

Please circle your primary method in the list above.

Question 9. How do you store results from bioinformatics programs that you have obtained in your research? (Tick those that apply)

- □ in a database
- □ on paper (printed)
- □ in a spreadsheet
- □ in a text file
- □ in a Word document
- □ other: ____________

Please circle your primary method in the list above.

Question 10. For each of these tools, indicate how you often you use them:

	Often	Sometimes	Never
Genbank	○	○	○
BLAST	○	○	○
Primer3	○	○	○
EMBOSS	○	○	○
ORF Finder	○	○	○
Repeat Masker	○	○	○
VecScreen	○	○	○
Other:_______________	○	○	○

Question 11. For each of these tools, indicate how well you can make use of them:

	Very well	Enough to be useful	Poorly	Don't know
Genbank	○	○	○	○
BLAST	○	○	○	○
Primer3	○	○	○	○
EMBOSS	○	○	○	○
ORF Finder	○	○	○	○
Repeat Masker	○	○	○	○
VecScreen	○	○	○	○
Sequencher	○	○	○	○

Question 12. Describe one example from the software that you have used that you found useful and easy to use:

Question 13. Describe one example from the software that you have used that you found frustrating to use:

Question 14. Please describe two tasks you perform with genetic sequences, using the following headings as a guide:

Goal:	The task you wish to perform
Inputs:	The information you enter into the tools
Outputs:	The resulting information you are after
Method:	The sequence of tools that you use
Problems:	Is there something about this task that you find frustrating or unnecessarily time consuming (if any)? Please explain.

Example:

Goal:	to design primers for a specific genetic sequence around a predicted intron location
Inputs:	the genetic sequence, the region to exclude (the intron location, we don't want a primer to bind to that region)
Outputs:	the forward and reverse primer sequences, their locations on the sequence, the calculated optimal temperature
Method:	we use BLAST to compare our cDNA sequence to a genomic sequence to infer intron locations, and then use Primer3 to design primers
Problems:	we don't know the intron locations, so we print the sequence out, draw where the introns are located and count the number of bases…

(A template is provided on the next page)

Task One

Goal:

Inputs:

Outputs:

Method:

Problems:

Task Two

Goal:

Inputs:

Outputs:

Method:

Problems:

Question 15. Please indicate if you would be willing to be approached to participate in further studies (e.g. trials of new user interfaces):

Name:

E-mail:

Phone Ext:

Please detach this question from the survey form and return separately.

Appendix B

Table 28 – Summary of applications examined in Chapter 5

Application Name	View Configuration	Interactions for Navigation
APIC (Bisson & Garreau, 1995)	One view 1. Multilevel (overview), Pan + Zoom	Pan by scroll bar, and buttons to move to next / previous feature or by a specified distance Zoom by buttons, labelled 'Zoom in' and 'Zoom out'
Apollo (Lewis et al., 2002)	One view 1. Multilevel (overview), Pan + Zoom	Pan by scroll bar, and buttons to move by a fixed amount Zoom by buttons to zoom in or out by increments
Artemis (Rutherford et al., 2000)	Two views, not interconnected 1. Multilevel (detail), Pan + Zoom 2. Multilevel (overview), Pan + Zoom	Pan by scroll bar Zoom by a vertical scroll bar
BLAST (McGinnis & Madden, 2004)	Two views, not interconnected 1. Detail view, Pan-only 2. Overview	Pan by scroll bar for Detail-view Zoom not available
ChARMView (Myers, Chen, & Troyanskaya, 2005)	One view 1. Multilevel (overview), Pan + Zoom	Pan by scroll bar Zoom by mouse modes for direct manipulation, Marquee Tool and Magnifying Glass Tool
DNAman (Woffelman, 2004)	Two views, not interconnected 1. Detail view, Pan-only 2. Multilevel (overview), Pan + Zoom	Pan in both views by scroll bar Zoom in Overview by buttons to zoom in or out
Ensembl (Hubbard et al., 2002)	Four interconnected views 1. Overview 2. Multilevel (intermediate) 3. Multilevel (intermediate) 4. Detail view	Pan by buttons to move by a fixed distance Zoom by buttons to zoom in or out by a fixed amount, and a select level control (see Figure 15).
GeneViTo (Vernikos, Gkogkas, Promponas, & Hamodrakas, 2003)	Two interconnected views 1. Detail view, Pan-only 2. Circular Map control	Pan by clicking on the circular map control (see Figure 16) Zoom not available
Genome Assembly Program (GAP) (Bonfield, Smith, & Staden, 1995)	"Template Display" One view 1. Multilevel (overview), Pan + Zoom	Pan by scroll bar Zoom by selecting mode "zoom in" or "zoom out" by button and selecting a marquee region to focus on
	"Join Editor" One view 1. Detail view, Pan-only	Pan by scroll bar and buttons Zoom not available
Genotator Browser (Harris, 1997)	"Model Display" One view 1. Multilevel (overview), Pan + Zoom	Pan by scroll bar, horizontal Zoom by slider, horizontal
	"Sequence Display" One view 1. Detail view, Pan + Zoom	Pan by scroll bar, vertical Zoom not available
Gestalt (Glusman & Lancet, 2000)	One view 1. Overview	Pan not available Zoom not available

Application Name	**View Configuration**	**Interactions for Navigation**
MEGA (Kumar, Nei, Dudley, & Tamura, 2008)	"Sequence Data Explorer" One view 1. Detail view, Pan-only	Pan by scroll bar Zoom not available
NCBI Map Viewer (Wheeler et al., 2005)	Two interconnected views 1. Overview 2. Multilevel (Intermediate), Pan + Zoom	Pan by button integrated into detail view or clicking on region in Overview Zoom by Select Level Control (see Figure 17) or clicking on region in Detail-view for context-menu with options to zoom by fixed amounts
NEBcutter (Vincze, Posfai, & Roberts, 2003)	"Overview" One view 1. Overview	Pan not available Zoom by marking location on map and clicking button, loads in Detail View
	"Detail View" One view 1. Detail view	Pan by buttons Zoom by button, returns to Overview
Primer3 WWW interface (Rozen & Skaletsky, 2000)	Two views, not interconnected 1. Overview 2. Detail view	Pan by scroll bar Zoom not available
RegulonDB (Salgado et al., 2001)	"Circular Map" One view 1. Overview	Pan not available Zoom by clicking on map, loads in Detail View
	"Detail View" One view 1. Detail, Pan-only	Pan by buttons, or by returning to the circular map and selecting a new location. Zoom by button, returns to Circular Map
SeqScape (Applied Biosystems, 2004)	Two interconnected views 1. Overview 2. Detail, Pan-only	Pan the Detail-view is panned by scroll bar or selecting a feature in the Overview Zoom not available
Sequencher (Gene Codes Corporation, 2003)	"Overview" One view 1. Overview	Pan not available Zoom by button, changes to Base View
	"Base View" One view 1. Detail view, Pan-only	Pan by scroll bar Zoom by button, changes to Overview
SeqVista (Hu, Frith, Niu, & Weng, 2003)	Two interconnected views 1. Overview 2. Detail, Pan-only	Pan the detailed-view is panned by scroll bar or selecting a feature in the overview Zoom not available
UCSC Browser (Karolchik et al., 2002)	Two interconnected views 1. Overview 2. Multilevel (intermediate), Pan + Zoom	Pan by buttons or selecting a location in the overview Zoom by buttons or specifying co-ordinates

Appendix C

Genetic Sequence Visualisation Study Recruitment Letter

You are invited to participate as a subject in a project entitled “A dynamic and continuously zoomable genetic sequence visualisation”.

We’re looking for participants for a usability study. In this study, you’d be working with a sample program intended for use by people who work with genetic sequences on computers. This session will require one hour of your time.

Our goal is to figure out how to make the interface more useful and user-friendly. You will perform some simple tasks with the program and comment on its ease of use and suitability. We are testing the program; we are not testing you!

We will record the computer screen during this session so that analysis can be made of the session later. The video will only viewed by the researchers and will be destroyed once the project is completed.

The results of the project may be published, but you may be assured of the complete confidentiality of data gathered in this investigation: the identity of participants will not be made public.

Does this sound like something you’d be interested doing? If so, please e-mail Paul Rutherford, who is carrying out the study as part of a Masters Thesis at Lincoln University, at **RutherP3@lincoln.ac.nz**

Contact Details

Paul Rutherford (Masters student, Lincoln University)
E-mail: RutherP3@lincoln.ac.nz Phone: 325-2811 ext 8376

Clare Churcher (Supervisor, Lincoln University)
E-mail: Churcher@lincoln.ac.nz Phone: 325-2811 ext 8905

John McCallum (Associate Supervisor, Crop & Food Research)
E-mail: McCallumJ@crop.cri.nz Phone: 325 6400 ext 3466

The project has been reviewed and approved by Lincoln University Human Ethics Committee.

Appendix D

Usability Study Consent Form

"A dynamic and continuously zoomable genetic sequence visualisation"

I have read and understood the description of the above-named project. On this basis I agree to participate as a subject in the project, and I consent to publication of the results of the project with the understanding that confidentiality will be preserved. I understand also that I may at any time withdraw from the project, including withdrawal of any information I have provided.

Name: ______________________________

Signed: ____________________ Date: ____________

Appendix E

Usability Study Script

I'm Paul Rutherford. This study is being performed as part of my Masters research at Lincoln University. We want to develop an application for working with data about genetic sequences that is easy for people like yourself to use. We've brought you here today to see what you think about some ideas that we are investigating. We want to see how well these ideas work for you, which don't work, and so on.

This evaluation should take about an hour.

We're going to be recording the screen of the computer you are working on. This is for analysis purposes only. It's so that I don't have to sit here and scribble notes and can concentrate on talking with you. It will be seen by only me and my supervisors. It is strictly for research purposes only.

Like I said, we'd like you to help us with a tool we're developing. It's in an early stage of development, so not everything you see is going to work right. This also means we aren't committed to anything yet, so any advice is really valuable because it means we don't have to spend time developing things that aren't going to be useful or effective.

Now I'd like to read to you what's called a statement of informed consent. It's a standard thing I read to everyone I interview. It sets out your rights as a person who is participating in this kind of research.

As a participant in this research:

- You may stop at any time.
- You may ask questions at any time.
- You may leave at any time.
- There is no deception involved.
- Your answers are kept confidential.

Any questions before we begin?

Let's start!

[Study One: Investigate Terms]

[Study Two: Evaluate Panning Controls]

Introduce computer-based tasks

Now we move onto the next part of the study. The procedure we're going to do here goes something like this: I'll show you part of an interface we are investigating, and you have a go at performing a few small tasks with it. Then I'll ask you a few questions about it, and we'll take a look at another part.

While you're using the computer I'd like you to say your thoughts aloud. That gives me an idea of what you're thinking when you're doing something. Just narrate what you're doing, sort of as a play-by-play, telling me what you are doing and why you're doing it.

Does that make sense? Any questions?

[At this point, screen recording was started].

Introducing new control

Here is a new control. Can you point out to me which parts look like you'd use them to move the sequence? What would you call each part?

After tasks with control

What were your thoughts on using that control? Did you like using it?

[Study Three: Investigate Symbols & Mappings]

[Study Four: Evaluate Zooming Controls]

[Study Five: Investigate Other Possibilities]

[Final Discussion]

How 'realistic' are the tasks?

What size genetic sequences do you usually work with?

How well do you think these controls would work for your sequences?

What do you think are the limitations of what you have seen here?

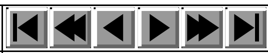
Is there anything else you'd like to say about what we've looked at today, or this study?

Thank you for your time!

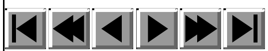
Appendix F

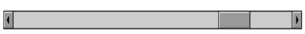
Task Satisfaction Questionnaire

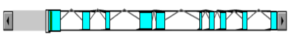
Of the controls you have just used, which performed best for each of the following tasks?

			
Finding a **feature** on the sequence	○	○	○
Moving to a **position** on the sequence	○	○	○
Browsing the **features** on the sequence	○	○	○
Overall, I prefer	○	○	○

Comments / Suggestions:



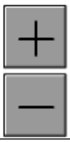
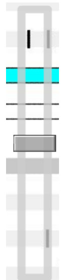




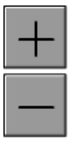
Other:

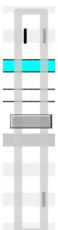
Task Satisfaction Questionnaire

Of the controls you have just used, which performed best for each of the following tasks?

			
Finding a **feature** on the sequence	○	○	○
Moving to a **position** on the sequence	○	○	○
Browsing the **features** on the sequence	○	○	○
Overall, I prefer	○	○	○

Comments:





Other:

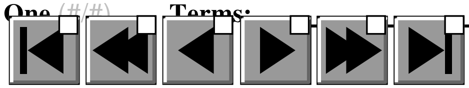
Appendix G: Usability Study Observer Sheet

Observer Sheet

Terms

Forward sequence	Panning
Reverse sequence	Zooming
Position / Location	Overview
Detail View	Features / Annotations

Study One (#/#) **Terms:** ____________________

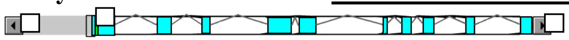


Noticed: Hold buttons / Inertia-Speed / Meaning ▶▶
Find Feature: Easy Difficult
Find Position: Easy Difficult
Browsing: Easy Difficult
Overall: Satisfied Not
Speed was: Too Fast / Too Slow / Okay

Study Two **Terms:** ____________________

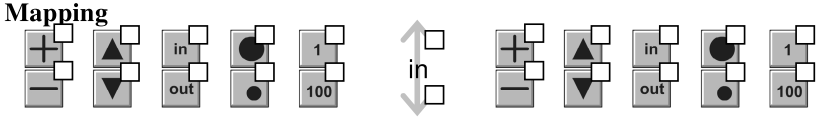
Find Feature: Easy Difficult
Find Position: Easy Difficult
Browsing: Easy Difficult
Overall: Satisfied Not
Speed was: Too Fast / Too Slow / Okay

Study Three **Terms:** ____________________

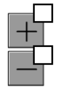


Saw **window**: Easily Difficulty
Find Feature: Easy Difficult
Find Position: Easy Difficult
Browsing: Easy Difficult
Overall: Satisfied Not
Speed was: Too Fast / Too Slow / Okay
Overview was: Lacking / Sufficient / Good

Mapping



Study Four (#/#) **Terms:** ____________________



Orientation: Correct Confusing
Find Feature: Easy Difficult
Find Position: Easy Difficult
Browsing: Zoom Pan only
Browsing: Easy Difficult
Overall: Satisfied Not

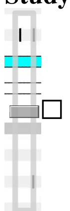
Panning ideas:

Study Five **Terms:** ____________________

Orientation: Correct Confusing
Find Feature: Easy Difficult
Find Position: Easy Difficult
Browsing: Zoom Pan only
Browsing: Easy Difficult
Overall: Satisfied Not

Panning ideas:

Study Six **Terms:** ____________________



Saw control: Easily Difficulty
Find Feature: Easy Difficult
Find Position: Easy Difficult
Browsing: Zoom Pan only
Browsing: Easy Difficult
Overall: Satisfied Not
Visibility was: Appropriate / Inappropriate

Panning ideas: