Support for Decision Making on the World Wide Web

A thesis presented in partial fulfillment of the requirements for the degree of

Masters of Information Science

in

Computer Science

at Massey University, Turitea Campus

Palmerston North

New Zealand.

Jaimee Alam

2004

Abstract

This research explores tool support for information retrieval and comparison of multiple pieces of information on the web. The study identifies the main goals users may have in mind when using the Internet in this way, and the necessary activities users complete to fulfill their goals. The main goals web users have are information search, entertainment and consumer to business transactions. The tasks users perform on the web to fulfill their goals include collecting, comparing, filtering and processing web information. These tasks form a decision-making cycle on the web and depending on the goal at hand, users may or may not necessarily undertake all the tasks or sub-steps in any sequential order.

Industry web support tools have been analyzed to find out how effective they are in supporting a common user's activities. These tools include web browsers (Netcaptor Browser and Internet Explorer), editing tools (Notes Pilot and Edit Pad), plug-ins, research tools and window management systems. Both browsers are poor at arranging multiple windows and excellent at opening web sites. The Internet Explorer browser proved to be better than the Netcaptor browser at a number of activities including, selecting web content, copying web text and images and pasting web content into editing documents. When used with either browser, Microsoft Windows is good for arranging windows but poor in switching windows views, scrolling windows and re-sizing and re-positioning windows. Both editing tools are poor at re-positioning and formatting web content from an HTML environment to a text-based environment. The Notes Pilot tool is also poor in making calculations and returning to the browser. It is excellent at saving work and retrieving old files. The Edit Pad tool is successful at all other activities except re-positioning and formatting web content. It can be seen that tool support is lacking or current web-based tools support the user poorly in a number of areas.

The need for an integrated web support tool has been identified. The functional and non-functional requirements have been specified, the tool designed, implemented and evaluated by users. The users were requested to complete a questionnaire and conduct

a think-aloud walk-through session while completing three tasks using the integrated web support tool. The sessions were observed and results recorded. Most of the users strongly agreed with the proposition that the tool would be useful for personal or academic activities. The users recognized the tool's novelty, its efficiency, and also indicated an overall level of satisfaction. The users were less satisfied about referring back to web sites, getting the software to do exactly what they wanted and arranging the work space to meet their needs. Changes were made to the tool.

Acknowledgements

I would like to dedicate this thesis to my beloved parents, Late Dr Kazi Firoz Alam and Nazneen Ara Khanom. They have raised me to be the best I can be and never to take a day for granted. I thank them for supporting all the decisions I have made in my life.

I wish to thank my two supervisors, Associate Professor Elizabeth Kemp and Associate Professor Chris Phillips for all their dedication and contribution towards this thesis. Their direction has helped me shape this thesis into my best work to date and I am very grateful to the both of them. My sincere thanks goes to Mr Paul Lyons who assisted me throughout the programming part of Networker.

Table of Contents

Chapter 1: Introduction

1.1	Background	1
1.2	Goals of the Project	3
1.3	Method	4
1.4	Report Structure	5

Chapter 2: Information Retrieval and Comparison on the web

2.1	Information on the Web	7
2.2	Web users and tasks	9
2.3	Working with information	12
	2.3.1 Making choices on multiple items	15
2.4	Decision-making Process	17
2.5	Norman's model of interaction	20
2.6	Concluding remarks	22

Chapter 3: A Review of Web Support Tools

3.1	Web Browsers	23
	3.1.1 The navigational model of web browsers	23
	3.1.2 The components of a web browser	24
	3.1.3 Limitations on Web Browsers	29
	3.1.4 Browser Plug-Ins	29
3.2	Window Systems	35
	3.2.1 A Window Management System	39
3.3	Other Web based tools	41
3.4	Analysis of Current Web Support Tools	42
	3.4.1 Review of Web Support Tools	44
	3.4.2 Comparison between browsers and window systems	44
	3.4.3 Compatibility between Browsers and Editing Tools	48
	3.4.4 Comparison between editing tools	49
3.5	Concluding remarks	51

Chapter 4: Specification and Design of Prototype

4.1	Functional Specification	52
	4.1.1 Required features of the tool	52
4.2	Non-Functional Specification	54
	4.2.1. Required Non-Functional Features of the Tool	54
4.3	Design Issues	55
4.4	Snapshot of the prototype	58
4.5	Concluding Remarks	61

Chapter 5: Implementation of Prototype

5.1	Environment	62
5.2	Implementation Issues	62
5.3	Requirements Review	65
5.4	Screen snapshots	67
5.5	Concluding remarks	72

Chapter 6: Evaluation of Prototype

6.1	Tool Evaluation Techniques	73
6.2	Methodology	76
6.3	Results	80
6.4	Changes to the Prototype	84
6.5	Concluding remarks	87

Chapter 7: Conclusion

7.1	Discussion	88
7.2	Future Work	91

References 93

Appendices

Appendix A: Program Code 99

A.1 Driving Instructions 115

Appendix B: Evaluation Documentation 116

B.1 Description of Networker 116
B.2 Information Sheet 119
B.3 Questionnaire 120
B.4 Illustration of Networker at Work 122

Appendix C: Evaluation Results 128

C.1 Transcriptions 128
C.2 Questionnaire Analysis 132
C.3 Additional Comments 134
C.4 Participant Performance 137

List of Figures

Figure 2.1 How users scan a page 13
Figure 2.2 Users see the areas of the page that interest them 14
Figure 2.3 Multiple site Decision-making process on the Web 19
Figure 2.4 Norman's Interaction Diagram 20
Figure 2.5 Extension of Norman's Interaction Diagram 21
Figure 3.1 Toolbar for Internet Explorer 24
Figure 3.2 Toolbar for Netscape Navigator 25
Figure 3.3 Status bar 27
Figure 3.4 Tabbed browser interface 30
Figure 3.5 The EditPad Interface 32
Figure 3.6 Notes Page in Notes Pilot 33
Figure 3.7 Saving and editing a note in Notes Pilot 34
Figure 3.8 A report generated in Notes Pilot 34
Figure 3.9 Components of a WordPad window 36
Figure 3.10 Cascade Men in Task bar 40
Figure 3.11 Cascaded Windows 40
Figure 3.12 Vertical Tiled Windows 41
Figure 3.13 Assessment criteria for browsers and window systems 46
Figure 3.14 Assessment criteria for editing tools 50
Figure 4.1 Current Tool Support for the Decision-Making Framework 53
Figure 4.2 Proposed Tool Support for the Decision-Making Framework 54
Figure 4.3 One open web site 59
Figure 4.4 Two windows and WorkPad area in vertical form 59
Figure 4.5 Two web sites and a WorkPad area in horizontal form 60
Figure 4.6 Two web sites in horizontal form with hidden Work Pad 60
Figure 5.1 Networker with one browser window displayed 68
Figure 5.2 Supporting the Collect and Compare Phase 69
Figure 5.3 Supporting the Filter Phase 70
Figure 5.4 Supporting the Process Phase 71
Figure 6.1 Results table 77
Figure 6.2 The static appearance of Networker with the changes made (two browser windows and WorkPad) 86
Figure 6.3 The static appearance of Networker with the changes made (one browser window) 86
Figure B.1 Vertical layout of Networker tool using one browser window and workpad 118
Figure B.2 Horizontal layout of Networker tool using one browser window and WorkPad 118
Figure B.3 Login and type in URL 122
Figure B.4 Minimise workpad 123
Figure B.5 Transfer content to WorkPad 124
Figure B.6 Open web site 124
Figure B.7 Resized window 125
Figure B.8 Content transferred to WorkPad 125
Figure B.9 Content transferred to WorkPad 126

Figure B.10 Format and Analyse content in Workpad 126
Figure B.11 Final selected website 127

List of Tables

Table 3.1 Components of a Window 37
Table 3.2 A task sequence when a user is conducting the web decision-making process 45
Table 3.3 Comparison of two browsers and a window system 48
Table 3.4 Compatibility between browsers and editing tools 49
Table 3.5 Comparison of two web-editing tools, Notes Pilot and Edit Pad 50
Table 5.1 Review of Functional Requirements 66
Table 5.2 Review of Non-Functional Requirements 67
Table 6.1 Questionnaire on the Networker 79
Table 6.2 Evaluation Results 81
Table 6.3 Analysis of Results 81
Table 6.4 Positive responses from participants 83
Table 6.5 Negative responses from participants 83
Table B.1 Questionnaire 120
Table C.1 Question 1 132
Table C.2 Task A 137
Table C.3 Task B 138
Table C.4 Task C 139

Chapter 1: Introduction

The focus of this thesis is on information retrieval and comparison on the Web, and the tools that are available to effectively carry out these activities. The research identifies the key activities in web decision-making, and how current tools assist web users to complete these activities. A tool that addresses the features lacking in current web support tools is designed and prototyped. Useful insights are provided into an average user's activities on the Web.

1.1 Background

The World Wide Web is a hypertext system. A pointer from the main text to a related document is called a "link". Most documents on the Web contain links to navigate to other web pages. Users traverse the links using a pointing device, e.g. a mouse. The process of reading web pages and traversing links to more web pages is called browsing or surfing. Users may browse with serious goals in mind or for entertainment (Lehnert, 1998).

The Web is a vital information resource for users. It is the complete set of electronic documents residing on all Internet servers that use the HTTP protocol (Johnson, 2000). Users can access web sites i.e. electronic pages full of valuable information, in a consistent and readable format using web browser software. Users will only feel satisfied that their goals are fulfilled if the tasks are conducted in an orderly fashion, i.e. their goals are met efficiently. It is very critical that the information content is easily located and read, and is both reliable and recent so that it can be sourced.

There are about 655 million users on the Internet in the world according to the CIA's world fact book (ClickZ Stats, 2002). This may increase to 709.1 million in 2004 (Emarketer, 2004). The number of ISPs in 2002 was 24,666 according to the CIA World Fact book (2003). Nielsen//NetRatings tracks the number of active users for approximately 20 nations, which account for more than 90 percent of the world's

Internet users (Nielsen //NetRatings, 2003). Web pages are being added at a rate of 7.3 million pages a day worldwide.

Researchers at the Online Computer Library Center claim that the Web in 2001 contains 8.4 million unique sites, compared to 7.1 million in year 2000 (Pandia, 2001). The Web continues to remain predominantly an English language medium. 73 percent of the sites have material predominantly written in the English language. Out of the present 8.4 million web sites, 3.1 million are classified as public sites (i.e. a site that offers content that is freely accessible to the general public), 2.1 million are classified as private (with restricted access) and 3.2 million are classified as provisional or in a transitory or unfinished state according to the Online Computer Library Center (Pandia, 2001).

Although increased emphasis is being placed on the Web, little empirical research has been done on the effectiveness of this hypertext-publishing medium, its user interfaces (web browsers), and users' mental models of the Web and hypertext (Nahl and Paervaiz, 1997).

There are seven classifications of the different types of content available on web sites (Landsberger, 2001):

- personal with biographic data: to market oneself, often called "vanity pages";
- promotional: to sell a product;
- informational: to share information on a particular topic or hobby;
- persuasive: to convert people to a particular point of view;
- instructional: to teach a unit or course of study;
- registrational: to register for courses, information, and/or products, accumulate a database of, and simplify communication with, registrants;
- entertainment: to purchase concert tickets, play electronic games and watch movies etc.

Most importantly, it is difficult to understand the nature of the interaction between users and information resources. Access can be rapid; links might cross boundaries of location, subject matter, content form, style and interactivity in a way that has no real equivalence in the paper world (Vaughhan and Dillon, 1998). Studies show that the three critical tasks for web users are to choose, collect and compare information (Nielsen, 2001).

1.2 Goals of the Project

Users commonly make purchasing and other decisions after comparing information downloaded from multiple web sites. The value of this information increases when it is combined (Krug, 2000). Comparing multiple pieces of information defines an important web activity that deserves explicit support from web applications. Currently, standard web browsers and text editors provide lower level support; the former allows a user to download information, the latter allows a user to collate information, but a higher level of support would be ideal in a tool that integrates the two functions.

Specifically, the goals of this project are:

- To identify user goals on the Web and the tasks users complete to achieve these goals.
- To derive a process for decision-making on the Web.
- To analyze current tools and determine to what extent these tools support user tasks.
- To develop key functions for a web support tool that overcomes the shortcomings of the current tools and assist users to deal with multiple web sites.
- To design and implement a web support tool, and address future development areas.

1.3 Method

The approach adopted in this study is as follows:

- *Analyze the critical tasks for users on the web interface during information retrieval and comparison*

  This process begins with a review of expert opinions and scientific research on the pertinent tasks web users currently undertake. Different approaches used to collect, compare and choose multiple information on the Web are studied. This information is used to derive a comprehensive framework, based on the key tasks a web user conducts when making choices between multiple pieces of information. This framework is presented in the context of Norman's interaction model.

- *Review web browsers, window systems and web tools to support these activities*

  Two web browsers are discussed in detail, Internet Explorer and Netscape Navigator. Microsoft Window Systems and other supporting web tools that use similar window mechanisms are explored. The features of the web support tools, which support the user's decision-making cycle, are described.

- *Test the web support tools in a decision-making framework using three common web user tasks*

  Each web support tool is tested to find out how effectively it can support steps in the decision-making framework. These tests are conducted in the context of common web user tasks.

- *Analyze and summarize these results*

  Each web support tool is analyzed in the context of the decision-making process, and the tools are compared. The results are presented in tables and discussed.

- *Develop a tool specification from the results*

  A set of functional and non-functional requirements which define the key functionality of the new tool, is derived.

- *Design and develop a prototype of a web support tool*

  The tool is designed according to the stated requirements. Any functionality not implemented is also noted. A prototype is implemented using a standard industry integrated development environment, and documented.

- *Evaluate the prototype*

  Expert opinions and scientific research on the common software evaluation techniques are reviewed, and relevant techniques selected. Documentation is provided for the evaluation session. A number of web users are employed to test the prototype. Results are documented. The prototype is modified.

1.4 Report Structure

This report is structured as follows:

Chapter 1: Introduction

This chapter is an introduction to the report. It introduces the Web, scopes the problem, overviews the problem description, illustrates the method used to address the problems, and describes the report structure.

Chapter 2: Information Retrieval and Comparison on the Web

This chapter sheds light on the collection and comparison of information on the Web. Web Information is introduced, how users process information on the Web is discussed and a framework of how web users conduct decision-making is developed. This framework is put in place in a Human Computer Interaction (HCI) context.

Chapter 3: A Review of Web Support Tools

This chapter is a background literature review on the current components of a web browser. Also, the Microsoft Window Systems Manager and plug-in software are described. Current web based tools such as, editing tools and web document organizational tools are also explored. A comparison is made between these tools and the shortcomings of these tools are also examined.

Chapter 4: Specification and Design of Prototype

This chapter is concerned with the requirements specification and interface design of the prototype. Functional and non-functional requirements for the prototype are detailed. Different design approaches for the prototype are considered and a rationale behind the final design is given. Draft snapshots of the interface of the prototype are shown.

Chapter 5: Implementation of Prototype

This chapter reviews the implementation of a lo-fi prototype of the web based analysis tool. Implementation issues are discussed in detail. The requirements implemented are also discussed. Finally, screen shots of the completed tool are provided.

Chapter 6: Evaluation of Prototype

This chapter evaluates a prototype of the web based analysis tool. Background literature on user evaluation is provided. The methodology used for usability evaluation is presented. The user background and the results from the survey and usability tests are discussed. Common threads and differences in the results are reviewed. The minor changes made in the tool are listed.

Chapter 7: Conclusion

This chapter highlights the main conclusions of the thesis and discusses the opportunities for future work.

Chapter 2: Information Retrieval and Comparison on the Web

This chapter briefly introduces the way that users perceive information in a web context and how users use this information effectively to accomplish their goals in a timely manner. The main goals of a user are to collect, compare and choose from information. A web decision-making framework is introduced in the last section of this chapter.

2.1 Information on the Web

Tillman (2001) considered the difference between information in print and on the Net and suggested how users should look for information on the Net. Tillman (2001) also describes the publishing categories available on the Net:

- Vanity publishing i.e. A vanity work may be a very specific document that has information of great value but it has not been through the peer review process intrinsic to scholarship or it has not been disseminated by the trade publishing industry;
- Grey literature e.g. pamphlets, preprints, technical reports;
- Advertising and Public Relations;
- Multimedia Support e.g. merging sound and graphics with print.

Experts have identified a number of quality factors against which information on the web could be evaluated (Nielsen and Morkes, 1997; King, 2000; Brandt, 2000; Testa, 2001). In regard to Accuracy, the web page lists the author and institution that published the page and provides a way of contacting him/her. The purpose of the document is provided. The qualifications of the writer or author of the document are provided. In regard to Authority, the web page lists the author's credentials, place of publication, and the page domain e.g. .edu, .gov, .org, or .net. In terms of Objectivity,

the web page has goals and objectives that are met, displays neutral information with no bias from the author's opinions or influences of advertising, classifies the group of audience it was written for and provides accurate information with limited advertising. The objective content is stripped of marketese and information presented without exaggeration, subjective claims, or boasting. The factor of Currency refers to the fact that the web page is current and updated regularly and the links are also up-to-date. Finally, in terms of the Coverage factor, the information can be viewed properly- not limited to fees, browser technology, or software requirements. The content must be cited correctly, all images must be balanced with the text, links must be tested and complement the documents theme. For text only frames, optional browser viewing could be included. If the page requires special software the software should be easy to access. If there is a fee requirement, this information should be easily accessible.

In addition, King (2000) identifies the key indicators to check for quality information on the Net:

- ease of finding out the scope and criteria for inclusion that lets the user see whether there is a match with his/her needs;
- ease of identifying the authority of authors, currency, last update, content updated;
- stability of information i.e. can users rely on the web page staying there?;
- ease of use in terms of convenience, platform standards, translation in native language, or organisation and speed of connection.

Kaiser (2000) designed a set of criteria for web site evaluation. The main areas covered are the web site's Appearance, Structure and Navigation, Content, Usability, and Overall Design Goals. The Overall Design Goals ensure that the web site needs to be simple, clear, consistent, appropriate, appealing and usable.

Lynch and Palmiter (2001) stress the importance of presenting information. They state that designers should strive for:

- structure and content that support user tasks;

- easy access to critical content, with important material higher up on the web site;
- increased readability and minimum distracters;
- a logical grouping of controls;
- feedforward and feedback mechanisms- connectivity.

The quality of the web information may be evaluated using web usability guidelines. The Inspection method is a quick and informal evaluation process uncovering defects and bugs in systems. The guidelines review or checklist method is one of many methods that fall under the Inspection technique. It identifies potential problem areas and the outcome depends on the choice of guideline to inspect the system against (Nielsen, 1995; Lynch and Palmiter, 2002). The web sites and their information can be assessed against a number of usability guideline checklists developed by Nielsen (1999) and Spool et. al, (1999). These checklists consider such issues as the quality of the content, presentation of content, navigation, and the efficiency of web site.

2.2 Web Users and Tasks

In regard to web user background, Nielsen (1997b) states that critics strongly believe that web site designers are focussing the usability of web information especially towards "stupid" users. The tests that are conducted are on the average users. Expert users can work out and achieve their goals from learning the complex presentation of information in the web sites currently available. In his defence, Nielsen (1997b) wrote an electronic article titled 'Are Users Stupid?'. Expert users may be capable of learning and using a complex user interface but are users actually willing to do so? From his research he found that users want to 'get in, get out, and move on with their own tasks.' This shows that web users have the same broad objectives, to accomplish their tasks in the most efficient, reliable and cost-effective manner.

Comparably, a study on user centred assessment of two web browsers proved that users' experience with an interface does not imply that users have used all of its functions and options. Thus advanced users may be novices at some tasks. Moreover,

the number of hours of use of a browser does not imply that users have the ability to use browsers for a variety of routine tasks (Nahl and Paervaiz, 1997). A study conducted by Stanford Institute for Quantitative Study of Society (Gerhard, 2000) assessed what goals users may have on the web. The results are outlined below:

Information Search

The most widespread use of the Internet today is information search for products, travel, research, hobbies, and general information. Virtually all users interviewed responded that they engaged in one or more of these information-gathering activities.

Entertainment

A little over a third of all Internet users report using the Web to engage in entertainment such as computer games e.g. online chess, role games. Thus, the current Internet is also emerging as an entertainment utility.

Consumer to Business transactions

A small fraction of Internet users are engaged in purchasing, stock trading, online auctions, and e-banking, with only a quarter reporting they make purchases online and under 15 percent of the users doing any of the other transactional goals.

Users perform tasks to achieve goals. Marchionini (1989) develops a distinction between open and closed tasks. Closed tasks have a specific answer and often integrate sub goals. Open tasks are much more subject oriented and less specific. Browsing can be used as a method of fulfilling either open or closed tasks. Open and closed tasks have individual browsing strategies. Two browsing strategies mentioned are: Search browsing, which is a directed search where the goal is known, and General purpose browsing, which is consulting sources that have a high likelihood of items of interest. In addition, there is a third browsing strategy known as Serendipitous browsing i.e. purely random browsing. This browsing strategy provides a continuum to distinguish between browsing as a method of achieving a goal and open-ended browsing with no particular goal in mind (Cove et. al., 1988)

According to Spool et.al (1999), the main purpose for users to use web sites is to 'find useable information for decision making'. There are many approaches a user may undertake to achieve the purpose including, research, software downloads, surfs and information searches. In contrast, another classification of user tasks is described by Lynch and Palmiter (2001). These tasks include, Navigation/Information tasks e.g to find out more detail; Search tasks e.g to find information; Transaction Tasks e.g to register, order, purchase, time to buy, process to buy; and Communication Tasks e.g to change orders, and to ship orders.

The tasks users perform on the Web include collecting, finding, exploring and monitoring information. According to a recent critical incident analysis, users' most important web tasks involve collecting and comparing multiple pieces of information, usually so that they can make a choice. Morrison et al. (2001) reflect the importance of each task that users find on the Web: Collect: 71% of the users searched for multiple pieces of information. They are driven by a specific goal, but are not looking for one particular answer; Find: 25% of the users search for something specific; Explore: 2% of the users looking around without a specific goal; Monitor: 2% of the users repeatedly visiting the same web site to update information. Visits are triggered by routine behaviour rather than a particular goal. Therefore, Nielsen (2001) concludes that the majority of the time web users collect and find information and thus, users are almost always goal-driven (96% of the time in the Morrison et al. study). In the study, the primary reasons for the respondents' important use of the Web were to compare/choose information or to understand information. In regard to compare/choose, 51% of the users evaluated multiple products or answers to make a decision; acquire information, 25% of the users got facts, documents, found out about a product, or down loaded information. In regard to understanding information, 24% of the users gained understanding of some topic; this generally included locating facts or documents. In brief, the three Cs, collect, compare, and choose tasks, describe most of the Web's critical use. Thus, usability studies should test the 3 C's and simpler tasks. The tasks are general and used by all users for diverse types of web sites.

2.3 Working with Information

In a Web context, users will also be actively searching for links that are of interest to accomplish the goal-directed tasks (Krug, 2000) (See Figure 2.1). When people encounter a text, familiar or not, they begin to process it from bottom-up by developing inferences about it. No previous knowledge of the text is required. The scanning or skimming of information on the Web utilises the bottom-up technique. Vaughnan and Dillon (1998) believe that research such as mental models can be extended to interact with documents in a digital domain. Users look for summary material, key words, closing and opening phrase and read serially on a web site until they find the relevant trigger words they are looking for.

Users' self-perceptions as well as browser interface design contributes to success in performing routine tasks (Nahl and Paervaiz, 1997). Studies of how users read on the Web found that users do not actually read: instead, they scan the text (Nielsen and Morkes, 1997; Vaughnan and Dillon, 1998; Spool, et al, 1998; Brinck, T., et al, 2001). Scanning is a process whereby users skim read the contents of a page, looking for trigger words and key statements that help understand the purpose of the paragraph or the nature of the argument and thus, form a summary of the page.

> 'What they actually do most of the time (if we're lucky) is *glance* at each new page, scan *some* of the text, and click on the first link that catches their interest or vaguely resembles the thing they're looking for. There are usually large parts of the page that they don't even look at.' Krug (2000) (See Figure 2.2).

In a research study with Morkes, Nielsen (1997a) found that 79 percent of web users scan rather than read. A summary of the plausible reasons behind the above statistic is outlined below: reading from computer screens is tiring for the eyes and about 25 percent slower than reading from paper, modern life is hectic and people simply don't have time to work too hard for their information, the Web is a user-driven medium where users feel that to be productive they have to move on and click on things and instead of spending a lot of time on a single page, users move between many pages

and try to pick the most relevant and appealing segments of each. Usability increased by 27 percent after using objective language in the user observation tests.

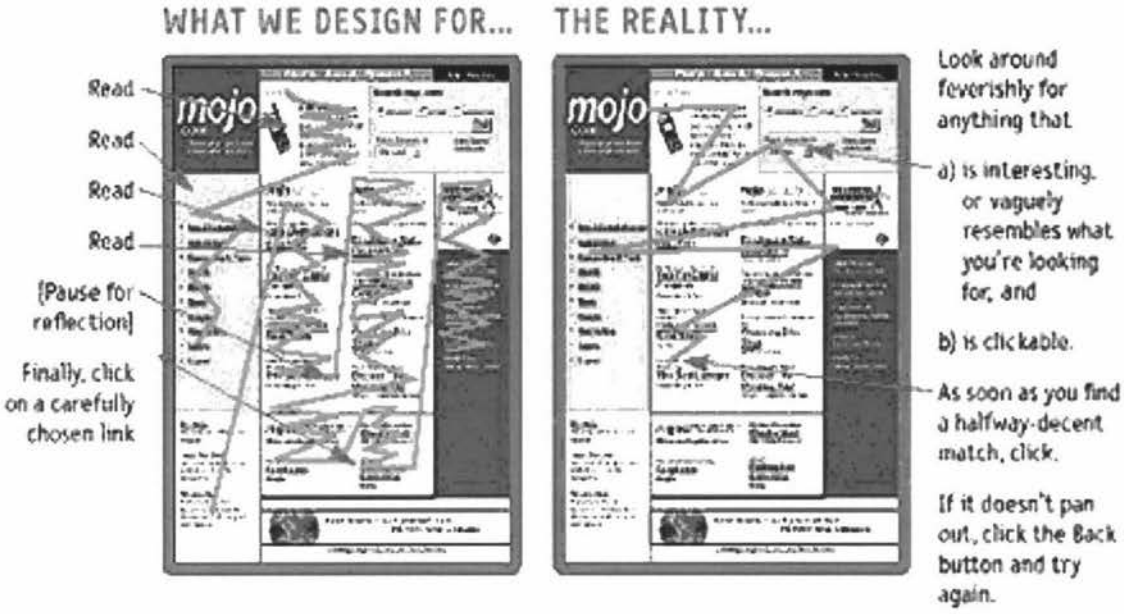

Figure 2.1: How users scan a page (Krug, 2000)

A number of experts suggested guidelines for presenting information to reach optimum scannability (Nielsen and Morkes, 1997; Spool, 1999; Brinck et al, 2001; Lynch and Palmiter, 2001):

- highlight key words and emphasise trigger words;
- use meaningful and enticing sub-headings and organise information by structuring articles with two or even three levels of headlines;
- bullet lists and use scannable layout;
- objective language, i.e. web content presented without promotional material, exaggeration, subjective claims, or boasting;
- use one idea per paragraph (users will skip over any additional ideas if they are not caught by the first few words in the paragraph);
- use active style of writing with concise text and begin paragraphs with the summary statement;
- use legible and clear fonts;

- use credible information, this can increase with the use of high-quality graphics and increase the use of outbound hyper text links;
- use hyper text to split up long information into multiple pages.

With the above factors, users can easily find more useful and relevant information and are able to scan the most information.

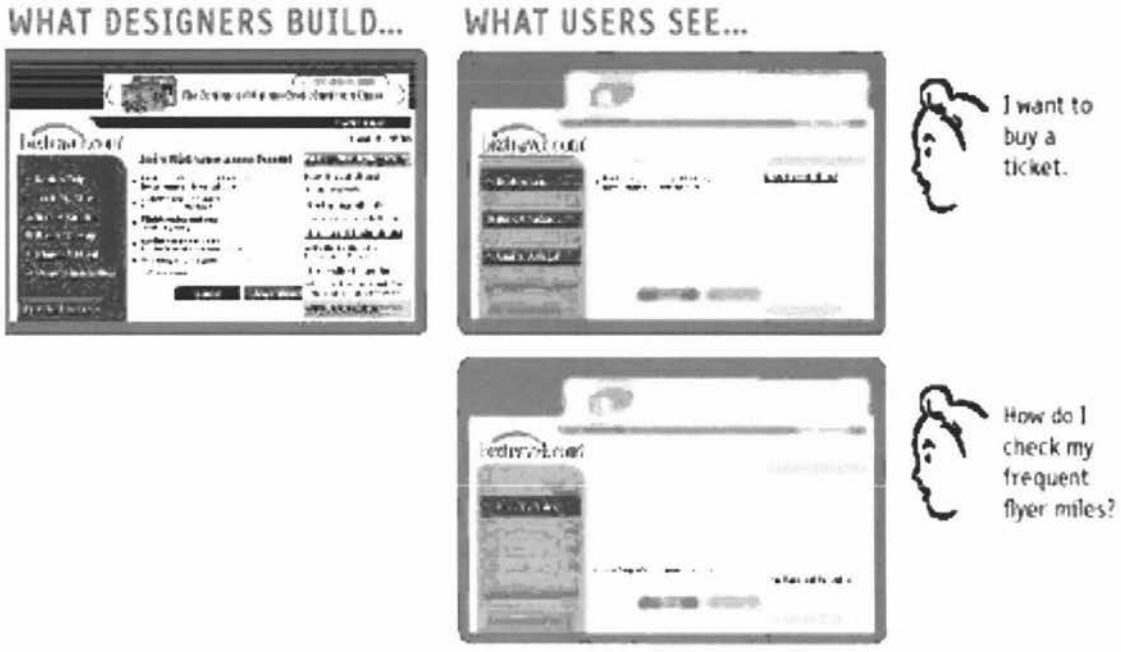

Figure 2.2: Users see the areas of a page that interest them (Krug, 2000)

One theory behind reading information on the Web is using the scanning technique. Another theory behind collecting information is that users may look for the "scent" of information. Scent extends the concept of navigation (Spool et al, 1998). Scent is how users find things. It helps users have a successful experience. Navigation is focused on the frame on the web page the user recently accessed, and the changes occurring after the manipulation of any interface widget. Scent, however, focuses on where the target information is and the changes, which occur when the site content is updated. Scent is a qualitative measure evaluated by user confidence and user satisfaction. Spool, et al (1998) believes that as user confidence increases, the scent gets stronger and thus, the likelihood of success increases. The design elements important to users before they click are: link quality, navigation, graphics, and information organization. Similarly,

the design element important to users after they click is seeing the desired information.

Spool et al. (1998) devised an evaluation criterion in connection with measuring how well web sites support the 'scenting of information'. In their guidelines they recommend that links should:

- be descriptive and predictable;
- match the page names;
- narrow the choices down from general to specific;
- be more content based and placed higher up on the web sites;
- be a minimum length to be useful.

Moreover, Spool et al. (1998) conclude that web sites should increase information levels within a single web site. Link organization, page layout and graphics is discussed further in their research. In the context of graphics, Spool et al. (1998) question whether graphics are helping users accomplish their goals. Moreover are the decorative graphics really worth the effort?

2.3.1 Making Choices on Multiple Items

The value of information increases when it is combined (Krug, 2000). Once information is found it needs to be organised, analysed, summarised, combined with other information and arranged in a form that is useful and understandable so that it sheds light into critical decision-making. Krug (2000) shares his ideas on the main tasks and the decision-making approach users conduct with information. He discusses the psychology and the decision-making approach users use while performing their goal-orientated tasks on the Web:

> 'Users don't make optimal choices. Users satisfice.'

Time is a precious resource. Users want to make the best decision in a very short time span. This is the reason why when users approach a potential web site, they skim read or scan the information. If it is relevant users read it word for word, otherwise they abandon the section altogether. Due to time constraints, users cannot choose the best option, they choose the first reasonable option from the information in front of them, and this strategy is known as satisficing. As soon as users find a link that seems attractive, there's a very good chance that they will click it (Krug, 2000).

Krug (2000) asks, why do web users not look for the best choice?

- Users are usually in a hurry;
- Users believe that there is not much of a penalty for guessing wrong;
- Users find weighing options may not improve the user's chances;
- Users may feel that guessing is more fun;
- Users may be pressed for time, and have low confidence to use the site.

Users of any technology do not like to read any form of instructions, help, manual, tutorials etc. Krug (2000) writes:

> ' We (users) don't figure out how things work. Instead, we (users) forge ahead and muddle through, making up our own vaguely plausible stories about what we're doing and why it works.'

If the information is not important to users, including expert users, they work through it in a disorganised manner, without taking the time to understand the order of the content. As soon as users find relevant information that will fulfil their task they may not even try to explore the most efficient way of completing the task, instead they may continue to complete their tasks using the recent information and the process they always use.

To reflect, the literature shows the lack of support currently available to web users looking for several answers or information about multiple products on the Web. Nielsen (2001) states that users handling content from multiple web sites are involved in Collect, Compare and Choose activities. Moreover, the value of information, as

Krug (2000) points out increases when it is combined. Krug (2000) also mentions that web users do not find an optimal choice, instead they find a satisficing choice. One possible approach for users to increase the chance of reaching to the best choice is to use a web support tool that can combine all the web choices onto one work area making the process of comparing and choosing options more efficient and effective and thus, leading to the optimal choice. It is important to note that Nielsen and Morkes (1997) believe that 79 percent of web users scan web content rather than read. This study is supported by Spool (1999), Krug (2000), Brinck et al (2001), and Lynch and Palmiter's (2001) research. According to these experts, web information must be presented so that users can reach the optimum scannability. Keeping this is mind, a tool is required that will allow web users to collect web information from multiple websites through the process of scanning and skimming and then compare the information using the tool to help enhance their decision-making process.

2.4 Decision-making Process

From an analysis of the literature, a decision-making framework has been designed, whereby the user begins by collecting web sites, and then moves onto comparing the web sites. Segments from each web site are filtered out for further analysis. Once a web site is selected then its content is processed further i.e. a product is ordered electronically or a service is booked for electronically. This cycle iterates until the user is satisfied with the end product.

The Web decision-making cycle has four steps namely, collect, compare, filter, and process (See Figure 2.3). In essence, it is a sequential life cycle, users first collect, then compare, filter and process web content. The four steps are described below:

Collect

This is a process wherein users gather individual web sites that may be relevant to the user's goals. Users need to browse from one web site to another and skim read each site to find useful content. These two sub processes, navigate and scan are correlated and together contribute to the collection of web sites. Navigation is a sub-process

wherein users move through or browse from one web site to another in order to find more details. Users may be moving from one section to the other following links on the same web site i.e. internal links, or moving from one web site to another ie external links or moving within a web site and between web sites. Skimming or scanning is a sub-process whereby users read serially until they come across key words, summaries, sentences, images, headings, or paragraphs that interest them. In the completion of the two sub-processes, users may find specific areas in web sites that interest them and the step is complete once the relevant information is recorded.

Compare

The second step in the decision-making framework deals with comparing the content of the collected web sites. Essentially it differentiates between useful and non-useful material. This step can be used whether there is one site or several sites present. A single site may have diverse products available for the user to compare amongst each other.

Filter

This is a step in the framework where a user selects the content from the web sites, transfers it to a common work space and makes further analysis until a decision is made.

Process

Essentially, it is the interaction between the web server and the user. The selected web site needs to be electronically processed in order for the user's goal to be met. This step is web site dependent and includes filling out order forms, verifying payment details and keeping a record of all the electronic documents processed by the web site server.

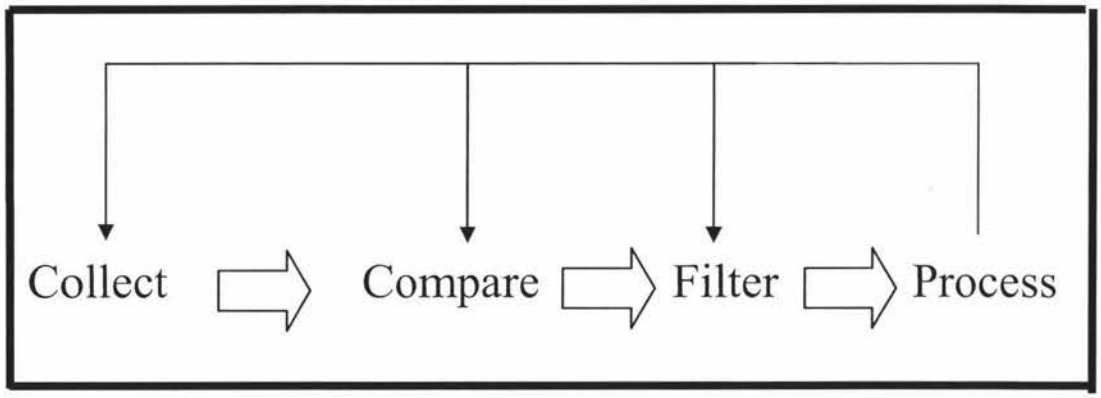

Figure 2.3: Multiple-site Decision making process on the Web

Once the user is satisfied, and the optimum decision is made, i.e. a user has purchased a ticket, has completed the assignment, then the user has satisficed the goal. Krug (2000) states that users do not figure things out, they muddle through. In reality, users may move backwards and forwards between the decision-making steps until the optimum decision is made and the task is complete. An example of this iteration is that the user may have made a choice while skimming through all the options and information. However, a certain subject area or key word may trigger new thoughts and the user may wish to research further into the topic. In this case, the user needs to go back to the Collect phase. The above framework considers all the routes and paths a user may travel through when making decisions for fulfilling various information search goals. It is understood that a user may not necessarily undertake all the activities or sub-steps in the process or conduct the activities in any order.

2.5 Norman's Model of Interaction

The purpose of an interactive system is to aid a user in accomplishing goals in an application domain. The user inputs information into the system and then the system processes and displays the results. The user analyses the information and once again inputs more information. This process is repeated until the user is 'satisficed' by the result and completes the task by keeping a record of the information.

Norman (1988) created a model of interaction that describes a cycle of activities that the user and system undertake (See Figure 2.4). There are two major phases: execution and evaluation.

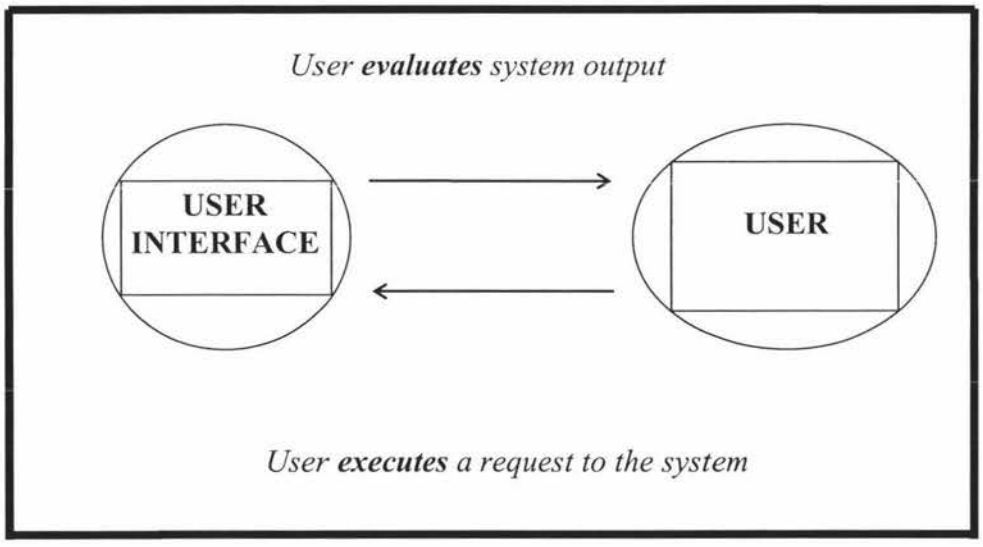

Figure 2.4: Norman's Interaction Diagram

The execution phase is divided into four activities the user conducts:

- establishing the goal;
- forming the intention;
- specifying the action sequence;
- executing the action.

The evaluation phase is divided into three activities the user conducts:

- perceiving the system state;
- interpreting the system state;
- evaluating the system state with respect to the goals and intentions.

Once the user evaluates the system's output the user must decide if the system state reflects his/her goal. If it does the interaction is successful, otherwise, the user must formulate a new goal and recommence the execution phase cycle (Norman, 1988). Norman's model underpins information retrieval and comparison on the Web. The user repeatedly requests information and evaluates the resulting system state. The decision-making framework in Figure 2.3 can be placed within Norman's model of Interaction. The Compare and Filter steps involve the evaluation phase of Norman's cycle. Similarly, the Collect, Filter and Process phase involve the execution phase of Norman's cycle The Filter phase is involved in both phases of the cycle because the user selects and filters the content from many web sites before executing a request to the system, and the user filters more content while evaluating the web sites (See Figure 2.5).

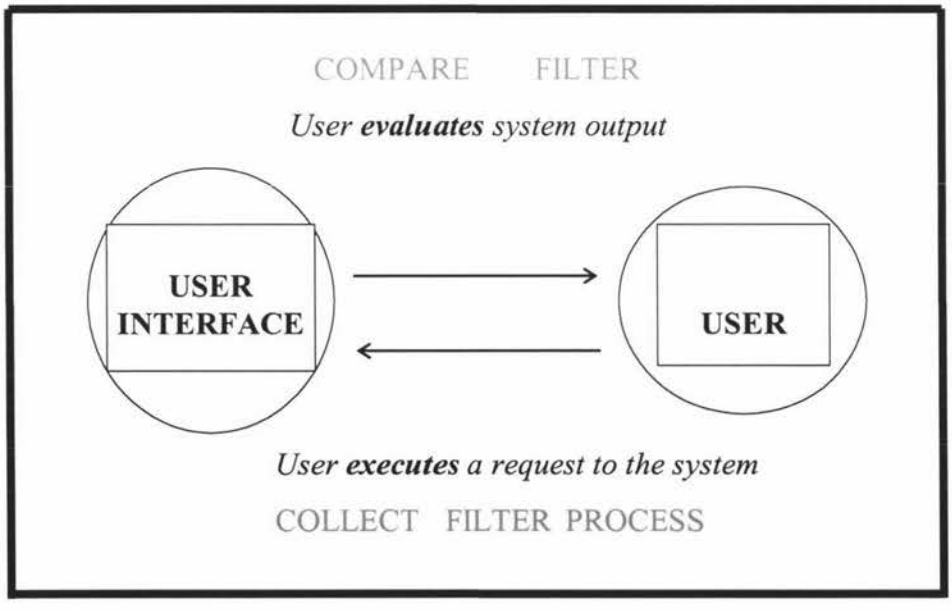

Figure 2.5: Extension of Norman's Interaction Diagram

2.6 Concluding Remarks

To sum up, this chapter demonstrates how users perceive information in a Web context and how this information is manipulated to accomplish user goals. There are three broad goals users pursue on the Web, Information Search, Entertainment and Consumer to Business transactions. Information Search is the main focus point for this study. In order to achieve this goal, users need to conduct numerous tasks and sub-tasks. The tasks include, collect and compare, filter and process content on the Web. A web decision-making framework is developed using the key tasks users conduct on the Web. Each of these broad tasks can be divided into sub-tasks right down to the physical keystrokes users make on the keyboard. Users prefer information content on the Web that is accurate, stable, relevant to needs, objective, current and has authority and coverage. Users, including experts and novices, want to easily find useful and relevant information and conduct their tasks in an efficient and cost-effective manner. Therefore, information content on the Web needs to be objective, concise and scannable. Users make satisficing choices on the Web. They prefer to "muddle through" or browse in a disorganised fashion to find the relevant material they require.

Now that information content on the Web has been discussed it is necessary to study whether browsers, window systems and web support tools provide appropriate functionality and support for information search tasks and whether the quality of information being browsed is organised and presented to aid goal orientated users.

Chapter 3: A Review of Web Support Tools

Chapter 2 has introduced a framework describing how web users collect, compare, filter and process information on the Web. Web support tools play an important role in supporting users to achieve their goals on the Web. In this chapter, current web support tools in the market will be reviewed; these include web browsers, window management systems, editing tools, plug-ins and other web-based tools.

3.1 Web Browsers

A web browser is the software program users use to access the World Wide Web, the graphical portion of the Internet (Lehnhert, 1998). Web browsers have evolved from textual interfaces to graphical and multimedia-based interfaces that focus more on integrated internet access designed to handle the most common Internet options. The point-and-click interface helped popularise the Web.

Although many different browsers are available, *Microsoft Internet Explorer* and *Netscape Navigator* are the two most popular ones. Both browsers are based on the first browser, NCSA Mosaic, developed at the National Center for Supercomputing Applications in the early 1990s (Lehnert, 1998). The Internet Explorer and Netscape Navigator browsers have more similarities than differences.

3.1.1 The Navigational Model of Web Browsers

Traditional World-Wide Web browsers, such as Mosaic and the various browsers it has inspired, use a depth-first navigational model. At any point in time, the user is "at" a particular node on the Web, having arrived there by following a path of nodes from some root. The user can choose to leave the current node either by following an outgoing link or by going back to the previous node in the path from the root. After

going back, the user can also choose to go forward to the next node on the most recent path from the root (Brown and Shillner, 1995).

Traditional browsers support other navigation methods in addition to these primitives; for instance, the user can jump to different URLs using the "Hotlist" and "Open URL" dialogs. Many browsers also offer multiple open windows, each with its own depth-first visitation stack. However, with the exception of Netscape and Internet Works, the Mosaic-like browsers such as Internet Explorer, are all single-threaded, so while one window is downloading a page, all of the windows become inactive (Brown and Shillner, 1995).

3.1.2 The Components of a Web Browser

The job of a web browser is to retrieve and display files, play media files and cater for email applications and related communication task inside the window. The content in each window will change as users move from one site to another, but each window has the same format. The majority of the browser space is filled with content that users read, links that users traverse and images users may view, hold, or print. The major components or parts in a browser are: Menu bar, Tool bar, Location or Address Bar, Directory Buttons, Content Area, Scroll Bar and the Status Bar (See Figures 3.1 and 3.2). They are provided by the browser to enhance navigability for users. Below is a description of the tool bar and other major components that make up a browser (Lehnhert, 1996 ; Ackerman, 1996).

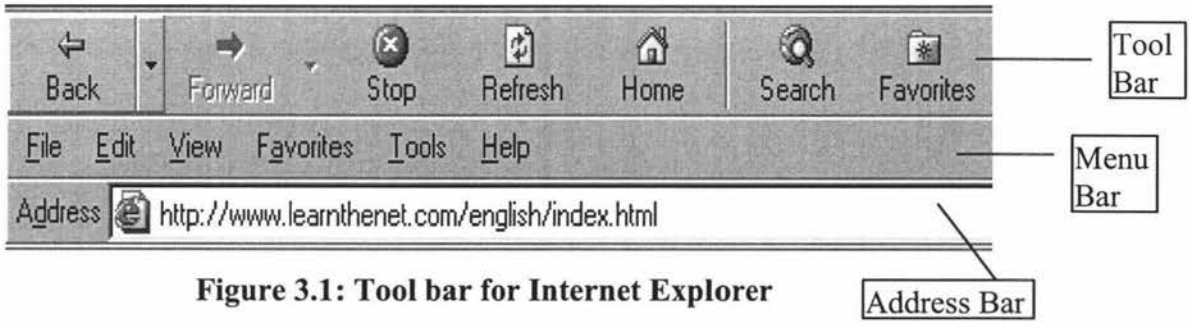

Figure 3.1: Tool bar for Internet Explorer

When a user first launches a web browser, the user double-clicks on the icon on the desktop, and then a predefined web page appears. This page is referred to as the home page or start page. With Navigator for instance, the user may be taken to Netscape's NetCenter or to a page selected by the Internet service provider. If the user wants to however, he/she can change the start page.

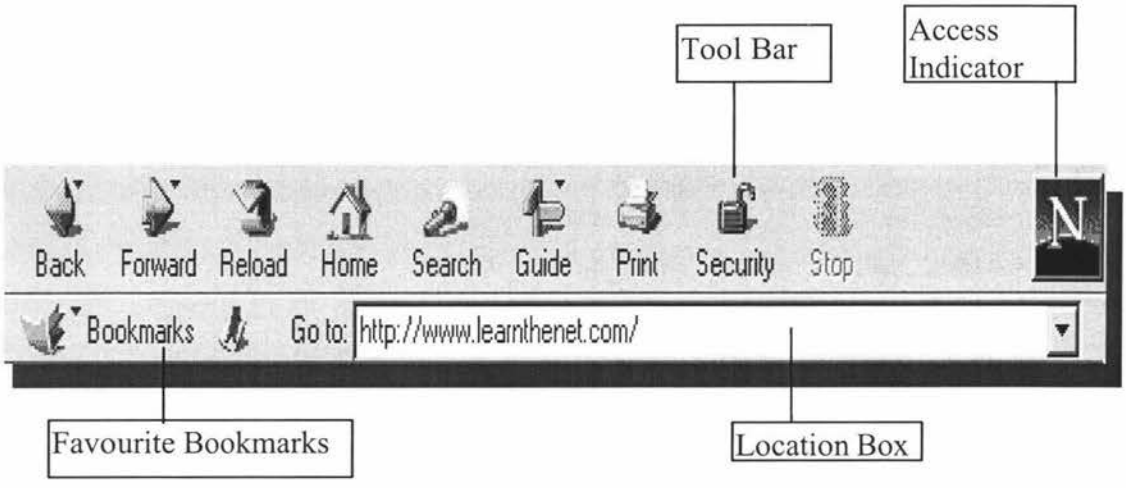

Figure 3.2: Tool bar for Netscape Navigator

The row of buttons at the top of a user's web browser, known as the tool bar, helps users travel through the web of possibilities, keeping track of where the user has been. Below is a description of the buttons in common to Netscape Navigator and Internet Explorer (Lehnhert, 1996 ; Ackerman, 1996):

- The *Back* button returns to the previous page a user visited. The back button retraces hyper-links one step at a time.
- Having gone back, the *Forward* button returns a user to the page they just came from.
- *Reload* or *Refresh* loads the web page again. Sometimes all of the elements of a web page have not loaded the first time, because the file transfer was interrupted. Also when a user downloads a web page, the data is cached, meaning it is stored temporarily in the computer's memory. The next time users want that page, instead of requesting the file from the web server, the web browser accesses it from the cache. By reloading the page, this timely data is updated from the web server.

- *Home* takes the user to whichever home page chosen.
- *Search* connects to directories and search tools on the Microsoft or Netscape web sites.
- *Print* lets users make a hard copy of the current page loaded in the user's browser.
- The *Stop* button halts the loading of the current page.
- *Bookmarks* or *Favourites* button lets users record the addresses of web sites they want to revisit. Once users add a URL to the user's list, they can return to that web page simply by clicking on the link in their list, instead of retyping the entire address. In Netscape Navigator, the button next to the Location Bar is the Bookmarks option (See Figure 3.2). In Internet Explorer this is presented in the tool bar as a Favourites button and as an option in the menu bar (See Figure 3.1). Both components directly access the Favourites management system so that the user can directly view and manipulate the web sites he/she regularly uses. The management system keeps a record of the URL. However, the user can access the bookmark by the default name or a name the user has set for the web site.

Other important components in a Browser are:

The Location Box or Address Bar

Just under the tool bar, is a box labelled "Location," "Go To," or "Address." This is where users enter the address of a web site they want to visit. By clicking the small button to the right of the Location box, users can find a drop-down list of the most recent web sites they have visited (See Figure 3.2).

The Menu Bar

Located along the top of the browser window, the menu bar offers a selection of things users can do with a web page, such as saving it to the hard drive or increasing the size of the text on a page. Many of the choices are provided as buttons on the tool bar (See Figure 3.2).

The Access Indicator

Both Navigator and Explorer have a small picture in the upper right-hand corner of the browser. When this image is animated, it means that the browser software, known as a *client,* is accessing data from a remote computer, called a *server.* The server can be located across town or on another continent. The browser downloads these remote files to the computer, then displays them on the screen. The speed of this process depends on a number of factors: the speed of the connection, the size of the files users are downloading, how busy the server is and the traffic on the Internet (See Figure 3.2).

The Status Bar

At the bottom of a user's web browser window is the *status bar*. Users can watch the progress of web page transactions, such as the address of the site they are contacting, whether the host computer has been contacted and the size and number of the files to be downloaded (See Figure 3.3).

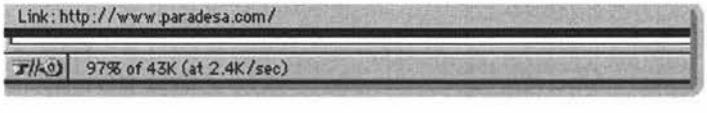

Figure 3.3: Status bar

The Scroll Bars

The vertical bar to the right of the browser lets users scroll down and up a web page. Users can place their arrow pointer on the up or down arrows and hold down their left mouse key. Users can also place the pointer on the slider control, hold down the left mouse key and drag the slider. If a web page is too wide to fit a user's screen, a horizontal scroll bar will appear at the bottom on a user's browser. This scroll bar works in the same way.

Directory Buttons

In the menu bar, Netscape Navigator has an option called 'directory'. This contains a list of interesting items on the Net users may want to explore. Several of the items can also be accessed by clicking on directory buttons placed directly underneath the Location Bar. Labels on the buttons are: What's New, What's Cool, Handbook, NetSearch and Net Directory. Similarly, Internet Explorer has a 'Tools' option in its menu bar that displays recent news, updates and chat rooms.

History Function

As users move from page to page and web site to web site, the browser remembers where they have been. Each time a user traverses a link from one document to another the browser updates a history list of documents. This list allows users to directly return to the documents previously visited. With Navigator, the user may select History from the drop-down list under the *Communicator* menu. Another way to move between pages with Navigator is by clicking the right mouse button. A pop-up menu will appear and users can choose to move forward or back. With *Internet Explorer*, users click the History button on the tool bar. Users can find a list of all the web pages they have visited during a specified period of time (See Figure 3.1).

3.1.3 Limitations on Web Browsers

Web browsers have some limitations. The lack of classification and evaluation mechanisms in hyper-linked documents on the World Wide Web, makes the role and value of digital documents difficult to assess (Nunberg, 1997). Navigation is one of the prime areas in which browsers lack support. Due to the restrictions in the navigation of Web browsers, Nielsen (1995) suggested a better bookmark management, and better diagram of navigation history. Also, pages from the same region of a site (that is, a sub-site) should be treated as a unit in history lists, bookmarks, and searches, etc. (Nielsen, 1997c). Users will then be able to access all the relevant pages in one step. Nielsen (2001) concluded that it is almost three times as important for users to collect multiple pieces of information than it is for them to collect pieces of information. Incidentally, browsers are mainly useful for helping users access single web sites but do not provide assistance for those users who want to obtain more than one answer. A browser can display one window at a time, and therefore, the browser needs to be re-opened each time a user wishes to display multiple pieces of information on multiple windows for the purpose of comparison. There are web site comparators available, which allow users to see the competitive price on products. These web sites compare content across multiple browsers; however, they are limited to comparing individual products against one or two criteria such as price and brand name. An example of such a site is Expedia. This is a travel web site that compares the fares across many airlines with the date and place of departure and arrival. However, these web site comparators can only compare a few pieces of information across many web sites and users may want to compare multiple pieces of information.

3.1.4 Browser Plug-Ins

Plug-Ins are support tools that users can download from the Internet. Some tools are available free of charge. While most tools have to be purchased, users can download a free trial of the software for a fixed duration. These tools are used to enhance the features of the current software available in an average computer-processing unit. The

downloaded software may be used for entertainment, education, work or research purposes. The software may increase processing speeds and be useful for multilingual and handicapped users. The tools described in this section will be examples of commercial window management tools.

NetCaptor

The NetCaptor (Netcaptor, 2003) software promotes itself as a separate Web browser. However, Komando (2000) thinks that it is more of an add-on for Internet Explorer as it requires that users have Internet Explorer installed first. The program relies on Internet Explorer for its basic browser functionality. NetCaptor organises multiple Web sites open in multiple browser windows (See Figure 3.4). It solves the problem of switching between browser windows by using a tabbed interface. Every new site visited will be the main site in the browser window; the previous sites will be displayed as little tabs at the bottom of the window. A click on a tab will send the user to that specific site.

Figure 3.4: Tabbed browser interface

Bhutan

Bhutan (University of Vienna, Department of Computer Science and Informatics Home page, 2002) is plug-in software that offers one main window and two support windows. This tool maps a navigational map in one window, the content in another and a navigational help window in a third window. The window in which a user calls up www.bhutan.at first is taken as the main window. It is also known as the content window. In this window the home page and the actual contents are indicated. The site map - a table of contents and a navigational help - are given in separate windows. This means that, depending on monitor size, the site map can stay open either beside or behind the contents window. It can be called up via the home page, by clicking on the coloured dot left top or the base line. Special program logic in more modern browsers prevents the site map from being loaded anew each time when called up, if it is already available on the window.

EditPad

Another plug-in, EditPad (EditPad, 2003), can open as many files at a time as needed. Users can switch between windows by clicking on tabs (See Figure 3.5). EditPad puts an icon in the taskbar that remains visible, even if EditPad is closed. This way a user has easy and fast access to EditPad, without the need to keep it running all the time. EditPad Pro is a text editor available for Internet Explorer users. If users work with many files at the same time, they will save time with the tabbed interface and the Project functionality for opening and saving sets of related files. The text editor has all the regular features like syntax colouring, bookmarks, auto indent, and paragraph symbols, etc. Many preferences can be made for each kind of file. File types can be text files, Java source code, HTML files or whatever users specify.

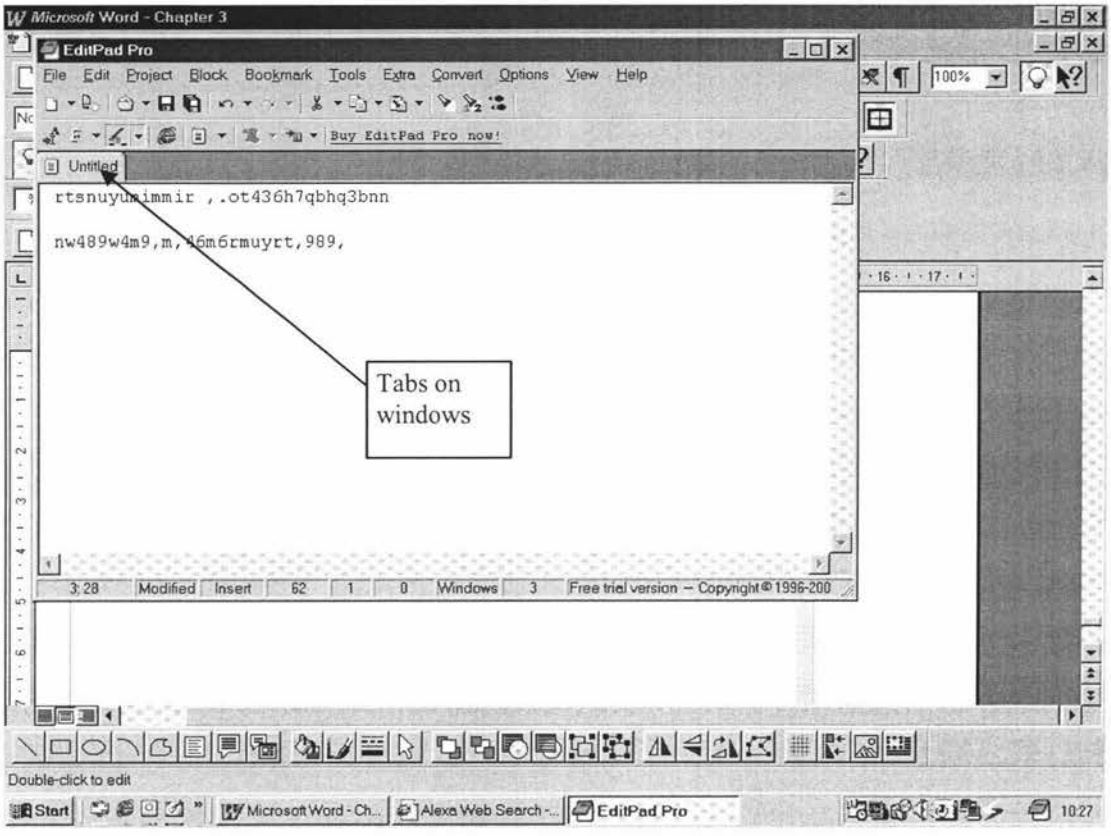

Figure 3.5: The Edit Pad Interface

Users can:

- drag text and images or copy-paste from emails, web sites;
- save HTML documents as textual documents;
- perform basic word processing functionality;
- integrate information from many web pages onto a text document.

However, Microsoft Word works better as a formatting and editing tool, as it has many drawing tools, additional toolbars, borders and other design widgets.

Notes Pilot

With *Notes Pilot (*NotesPilot, 2003) users can collect text and images from web pages. Following this they can drag 'n' drop selected text and images on the program icon and save it. Text can be collected from Outlook, Microsoft and the Internet Explorer

browser. Notes Pilot is a notepad with tree-like structure support, and can be accessed from any Windows application. Users select information and drag and drop it into Notes Pilot (See Figure 3.6). Notes Pilot is similar to the 'favourites' toolbar as it opens beside a web page and users can drag and drop web page information from this tool bar 'management' organiser. The shortcut icon to Notes Pilot can be found in the explorer bar, browser toolbar, browser menu, and on the desktop. In the start menu it works as software that saves web pages individually into folders as separate HTML pages (See Figure 3.7). Finally, Notes Pilot generates a report (See Figure 3.8) of all the pages in the folder, integrated onto one page, with each section headed by the URL of the corresponding web site.

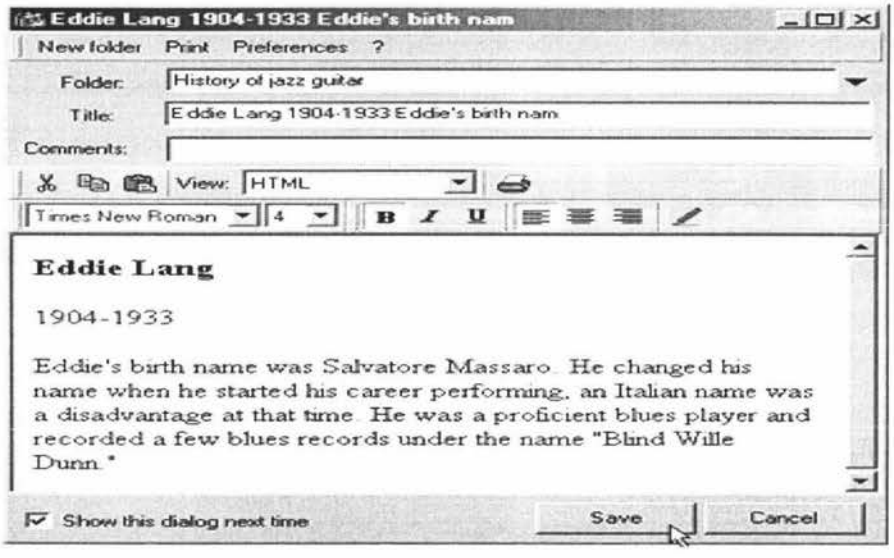

Figure 3.6 Notes Page in Notes Pilot

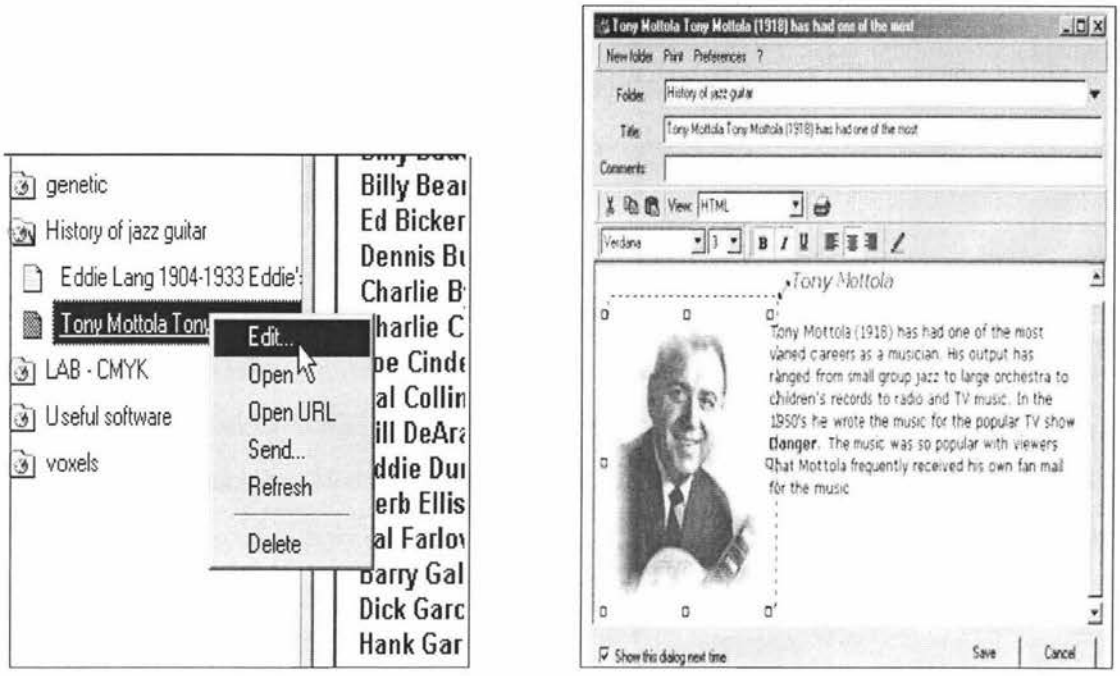

Figure 3.7 Saving and editing a note in Notes Pilot

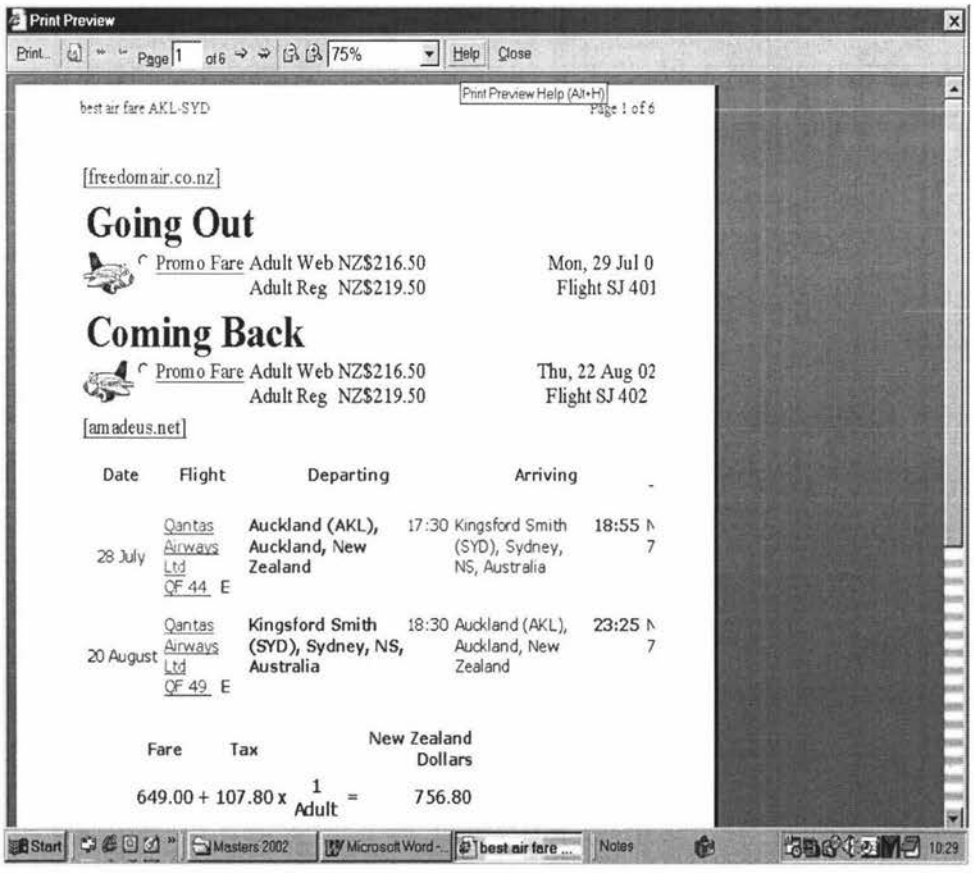

Figure 3.8 A Report generated in Notes Pilot

Alexa

Alexa (Alexa, 2003) allows users to surf more efficiently by using searches across related links and web sites as well as conducts all of the searches from one interface. Moreover, users can get critical web site information such as traffic rankings, contact information, site statistics and ratings. Alexa's version 1.0 tool bar was first released in September of 1997. The tool bar featured Site Information, Related Links, access to archived pages, access to Encyclopaedia Britannica content, and an instant messaging feature. In 1998, Netscape included the shortcut for Alexa into the 4.0 version of the Netscape Navigator browser as the centrepiece of the "Smart Browsing" feature set.

3.2 Window Systems

A window is a rectangular frame that displays a view of an object or a segment of an object and that provides access to controls by which users can manipulate the window view. Browsers use window systems to view and manipulate the web content (Figure 3.9). Most document windows contain *controls*, which are screen images that the user manipulates to control the display or the behaviour of the application (Windows Development Manual, 1998). The operations allow users to (Card et al, 1984; Myers, 1988, Bass and Coutaz, 1991; Shneiderman, 1998a; Dix and Finlay, 1998):

- create and delete windows;
- open and close windows;
- move windows on the display;
- bring a window to the top;
- resize windows;
- scroll a window;
- hide and show windows;
- zoom windows;
- change the view of the object on the window.

Most of the appearance and behaviour of the screen is controlled by a complex program called a *window manager*. The Window Manager creates and displays a new window when the user creates a new document or opens an existing document. When the user closes a window, the Window Manager is used to remove the window from the screen.

The window system translates user actions into events for the client program. The system accepts requests for output according to returned results of operations input in the form of events (Bass and Coutaz, 1991). Events are messages that describe user actions and report on the processing status of a user's application. When the system receives an event, it responds according to the kind of window that is currently active and where the event occurred (Windows Developer Manual, 1998). The interface objects on the window are illustrated in Figure 3.9 and tabularised in Table 3.1.

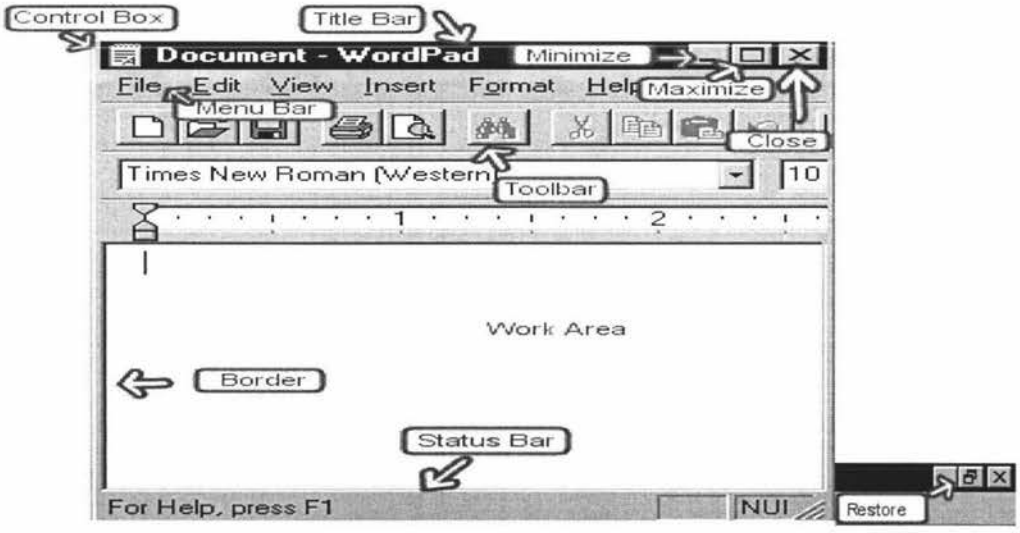

Figure 3.9: Components of a WordPad window (Baycon Group, 1999)

Table 3.1 Components of a window (Shneidermann, 1998a; Windows Developer Manual, 1998; Baycon Group, 1999)

Control Box	The control box provides a menu that enables users to restore, move, size, minimise, maximise, or close a window.
Scroll Bar (not shown in diagram)	Most windows have content that spreads over more than one page and thus, a scroll bar is used to move over the content of interest in the window. The basic operations of a scroll box are to move up or down and left or right.
Border	The border separates the window from the desktop. Most windows have a thick border or frame to accommodate selection and resizing or to distinguish windows from the background. Users resize the window by dragging its borders outward to expand it and inward to contract it.
Title bar	The title bar displays the name of the current file and the name of the current program. The title is highlighted when the window is active.
Minimise button	Users use the Minimise button to temporarily remove the window from view. While a window is minimised, its title will appear on the taskbar.
Maximise button	Users click on the Maximise button and the window will fill the screen.
Restore button	After a user maximises a window, clicking on the Restore button, will return the window to its former size.
Close button	Users click on the Close button to exit the window and close the program.
Menu bar	The menu bar displays the program menu. Users send commands to the program by using the menu.
Tool bars	Tool bars are generally display right below the menu, although they can be dragged and displayed along any of the window borders. Users use the icons on the tool bars to send commands to the program.
Work area	The work area is located in the centre of the window. Users perform most of their work in the work area.

Windows provide the possibility for several virtual screens to be visible simultaneously. However, in most systems only one window is connected to the keyboard at any time. This is described as the Active Window. There are numerous means to display virtual screens on the desktop. Windows can be overlapped, tiled, cascaded or split.

Overlapped Windows

With overlapped windows, a large number of windows can be opened at one time and users can compare objects in multiple windows. The user manually manages the size and location of multiple windows. Users may see multiple windows on a display with the appearance of partial or arbitrary overlaps. Windows may be moved at any point in the display and a portion of the window may be off the display. When overlapping windows are used, users need a mechanism to bring forward and make active a window that is totally or partially obscured by other windows. *Cascading windows* are overlapped windows laid one on top of the other. Designers have applied the familiar deck-of-cards metaphor by positioning a sequence of windows on the desktop from the upper left corner down to the lower right corner. Successive windows are offset below to the right to allow each window title to remain visible. Microsoft windows manager automatically lays out successively opened windows in the cascade form. Cascading windows is a way of organising a user's windows on the desktop (Baycon Group, 1999).

Tiled Windows

Tiling windows is another way of organising windows on screen (Baycon Group, 1999). In contrast to overlapping windows, the system manages the size and location of the tiled windows. Again, users can compare objects on multiple windows. The windows may be placed in fixed columns. Microsoft Windows Manager offers horizontal and vertical tiled columned windows. Using this technique, the screen space is utilised to the full, and since no window is obscured, the user never has to rearrange the screen to see hidden windows. Re-sizing of tiled windows causes reformatting and alignment of text according to the tile format.

Split Windows

Many word-processing applications, such as Microsoft Word, enable users to split their display to show two or more sections of a document, or two or more documents. The split displays can be made horizontal or vertical to allow side-by-side comparisons.

Window placement can be changed as a result of any of the following commands (Shneiderman, 1998b): move to top, move to bottom, reposition on display, change size and iconify.

3.2.1 A Window Management System

The concept of working in several windows at the same time is a fundamental aspect of Windows 98 (Windows 98, 1999). It is a rare occurrence, in fact, if users are not working with at least two windows simultaneously. To keep the windows organized, users have several options. To make a particular window active, users can either click on it, click on its button on the taskbar, or they can use the keys on their keyboard to switch to that window. If many windows are open and the user would like to make them uniform on the screen, they can right-click on the taskbar to organize the multiple windows. A menu bar opens to show all the options available to organise the windows (See Figure 3.10). If the user chooses the Cascade option from the shortcut menu, Windows 98 will organize the open windows into a "index card" style, stacking one behind the other, with all windows the same size (See Figure 3.11). The other option is to tile the windows. Windows 98 will place the windows right next to each other, with no overlap, either across the screen (Tile Horizontally) or up and down the screen (Tile Vertically) (See Figure 3.12). The interesting aspect is that the cascade menu is available on the task bar of Internet Explorer and therefore by default multiple browser windows can also be organised by cascade, or horizontal or vertical tiles.

However, the more windows a user needs to tile, the less readable each window's contents become as all the windows will be accumulated and arranged to fit onto the one screen. In the horizontal tile format, a small visible area of the screen is shown and the horizontal scroll bar placed in the bottom of the window, similarly, in the

vertical tile format a small visible area of the screen is shown and this is followed by the vertical scroll bar placed in the right side of the window. The last option that is selected from the menu e.g cascade, horizontal or tile window will be the next default arrangement for the windows the next time a user opens multiple windows. This window arrangement will not change until the user selects a different option from the task bar menu.

Figure 3.10 Cascade Menu in Task Bar

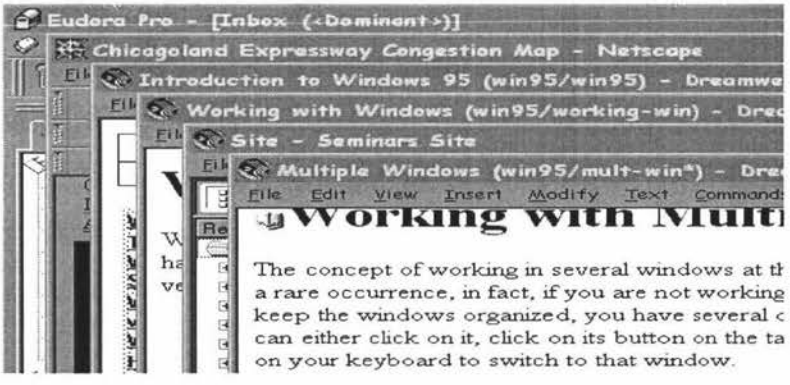

Figure 3.11 Cascaded Windows

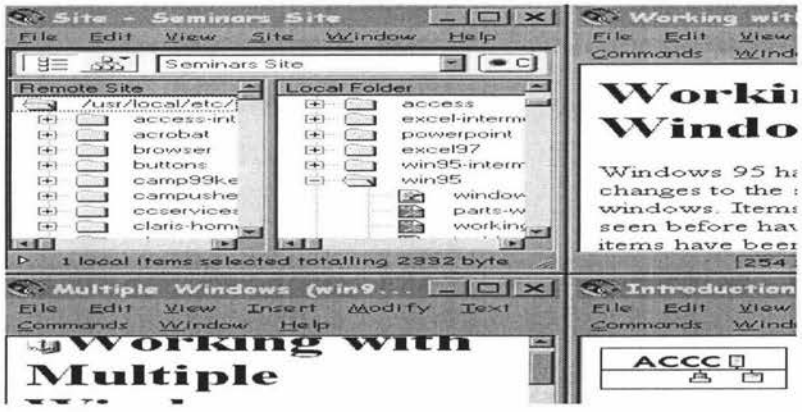

Figure 3.12 Vertical Tiled Windows

3.3 Other Web based Tools

Shneiderman and Kandogan (1997) developed a tool called the Personal Role Manager. Used instead of a window manager, it improves performance and reduces distraction while the user is working in a given role, and facilitates shifting of attention from one role to another. The key is to organise information according to the roles that an individual has in an organisation (Schneiderman and Kandogan, 1997). Tasks are hierarchically organised into subtasks via an outlining tool. An approach for window management in this tool is Elastic windows (Shneiderman, 1998b). The windows are organised in a space filling, hierarchically nested, tiled layout that was designed to support the personal role manager's requirement. Actions such as open or close can be applied to a group of windows to open up to six related e-mail messages. Multiple window operations were shown to enable faster task switching and structuring of the work environment, by opening, closing, or changing the size of 10 to 20 windows at a time. Users can change the layout according to tasks by applying operations on groups of windows i.e. container windows that surround member windows. When re-sizing, the system enlarges a window or group of windows while reducing the size of other windows in a group. A sub hierarchy of windows can be collapsed into a single icon or textual item.

Apperley et al (1999) argued that the cut-copy-paste functionality in GUI interfaces is limited in its functionality in circumstances where several related pieces of data needs to be copied from unstructured to structured set of data fields. To deal with this problem they created the "Stretchable Selection Tool" (SST). The conventional copy/paste sequence involves overlapping windows, a lot more mouse movement between multiple windows and users remembering the place to transfer data if the destination window cannot be kept visible. This tool is used when users undertake a goal directed search process. SST makes it clear to the user what action is pending. A visible band field is connected from the mouse cursor to the destination field, and a short text prompt is provided below the band. Thus, the user does not need to shift his/her mouse pointer, or attention between windows. In the situation where the source and destination fields of transfer are close together, the user left-click, drags, and right-clicks on successive pieces of data. Otherwise, The SST permits transparent access to the normal browser controls. When the destination page is out of sight, a pipe from the left edge of the screen is created that attaches itself to the destination component and prompts for information. The text on the pipe (SST band) updates to show the name of the target field, and provides a direction even when the destination page is not fully visible. Other means of using the SST is for copy pasting URLs. The SST provides a reduction in navigation overhead.

Additional web-based tools include, language translation tools, research assistance tools, document search tools, tools that can zoom-in or out, tools that disable graphics and tools that make browsing faster. There are also tools that can provide easy access to external links and references to web sites, and tools that format and rearrange multiple windows. These tools can be purchased commercially and installed into the browser or down loaded for free from the World Wide Web (Komando, 2000).

3.4 Analysis of Current Web Support Tools

The web tools described in section 3.2 are tools used to enhance a web user's decision-making process. A typical task sequence when a user is performing the web decision-making process is shown in Table 3.2. Users can collect web content via a

web browser such as Internet Explorer and then arrange the browser windows using a Window Systems Management tool or Netcaptor. Once the content is compared from a selected number of browser windows, the critical content can be copied into an editing tool such as Notes Pilot or Edit Pad. This process can be applied to different tasks such as booking an airline ticket and purchasing a television. An example of the airline scenario using the task sequence in Table 3.2 is as follows:

Task Sequence to purchase an airline ticket

Find the cheapest airfare from Auckland to Sydney departing on 25th July and returning on 27th July. Include flying points registration.

- Open up Internet Explorer
- Go to four different airline web sites
- Enter arrival and departure details
- Search options
- Do the same for all the others
- Compare all four web sites with content, navigation and appearance checklists
- Filter out the cheapest web sites that best meet the criteria
- Book a ticket
- Register for mileage points

An example of the scenario to purchase a television is as follows:

Task Sequence to purchase a television

Purchase a 54" SONY television ideally with up-to-date features, Dolby stereo system and a matching entertainment unit alongside. A budget of $US 5000 is available, only purchase by International Visa. Look out for additional perks e.g. free gifts, interest free terms.

- Open up Internet Explorer
- Go to four different electronics web sites
- Place price or browse price for SONY 54" brand
- Search options
- Do the same for all the remaining three web sites
- Compare all four web sites with content, navigation and appearance checklists
- Filter out web site that best meet the criteria
- With additional perks and International Visa
- Order television

3.4.1. Review of Web Support Tools

Two groups of web support tools, which are most relevant to the decision-making framework, are reviewed below. The first group consists of two browsers: Internet Explorer and Netcaptor, plus the Microsoft Window Management System. Netscape Navigator has not been used for the purpose of this test, mainly because it is not compatible with the editing tools. Instead, Netcaptor tool's browsing capabilities are tested against Internet Explorer. The second group consists of two editing tools, Notes Pilot and Edit Pad. It is necessary to compare the tools and find the shortcomings of these tools to develop the proposed tool requirements.

3.4.2 Comparison between Browsers and Window Systems

Table 3.3 highlights the strengths and weaknesses of the browser and window management tools when used to carry out the tasks previously described. Internet Explorer and Netcaptor use the Microsoft Windows Management system to arrange the web sites in a comparable form. The criteria used to evaluate the browsers and a window system are shown in Figure 3.13.

Table 3.2: A task sequence when a user is conducting the web decision-making process

Phases of the web decision-making framework	User Tasks
COLLECT	Open a web site.
	Scan the material.
	Navigate the internal and external links in the web site.
	Assess the quality and relevance of the content
	If the content is relevant then retain the window
	Iterate the above tasks until two or more suitable web sites are found.
COMPARE	Open windows via. the Internet Explorer Window Manager.
	Display windows in a tiled format.
	Assess the quality and suitability of the content between all the windows.
FILTER	Select and drag the sections of the web site that are needed for analysis.
	Analyse this information on one work space
	Select segments of a web site or a whole web site.

Phases of the web decision-making framework continued..	User Tasks
PROCESS	Return to the Internet Explorer Window Manager and focus on the web site selected.
	Order the product or process the bookings.
	Fill out the electronic forms with the user and payment details.
	System validates the information.
	Once validated, print the invoice or confirmation document.
	User may wish to keep a record of the electronic documents, specially the product description, user manual, guarantee, booking rules and restrictions, contact information and delivery instructions.

Assessment criteria for browsers and window systems

Poor Support

This category is selected for the tools that do not fully support the tasks such as integrating activities between browser and tool. This may mean that the user can not achieve the task in one single step or the tool is not sufficiently flexible to support the task. In some cases the user takes extra steps and it takes much longer to achieve the task with the support of the tool. Some tools make it difficult for the user to view the document and require scrolling in both directions. Other tools are unable to convert HTML to a readable text format.

Good Support

This category is selected for tools that support many steps of the task at hand. The tool may have limited menu options or commands, which restricts the user from completing the entire task. For example the Window Systems Manager in the Netcaptor tool arranges multiple windows. There are three formats available, however, only one is useful for comparing windows.

Excellent Support

This category is selected for tools that not only complete the entire task but in addition, allow the user a flexible environment or different menu options to complete the task. An example is the tabbed window in Netcaptor, which allows the user to switch between windows by the use of tabs.

Figure 3.13 Assessment criteria for browsers and window systems

Table 3.3 clearly shows the specific areas of support each tool provides. For instance, the browsers are used to open and arrange multiple web sites and later used to select and copy and paste web content into the recording and analysis software. Both browsers are poor at arranging multiple windows and excellent at opening web sites. The Internet Explorer Browser proved to be better than the Netcaptor browser at a number of activities including, selecting web content, copying web text and images and pasting web content into editing documents. When used with either browser, Microsoft Windows is good for arranging windows but poor in switching windows views, scrolling windows and re-sizing and re-positioning windows.

When the user views a single web site the web site is clear and readable. The readability of a web site drops when multiple windows are accessed. Every web site window has browser menus, buttons, scroll bars, frames, title and URL bars, and tool bars. The window space allotted by the system is used up mostly by these overheads leaving little readable web site content. In turn, as the number of windows increase, the allotted segment of desktop space for each window decreases in proportion. Consequently, readers can see less of the important web content and the content they can see may be difficult to read.

Table 3.3: Comparison of two browsers and a window system

Tools / Tasks	Internet Explorer			Netcaptor			Window Systems		
	Exc	Good	Poor	Exc	Good	Poor	Exc	Good	Poor
Open web site(s)	*			*					
Arrange multiple windows			*			*		*	
Scroll									*
switch from minimize to maximize view									*
reposition or resize windows									*
Select content from one web site	*				*				
Copy web text and images	*				*				
Paste into a documents (s)	*				*				

3.4.3 Compatibility between Browsers and Editing Tools

Table 3.4 highlights the integration the browsers have with the two web-editing tools. Internet Explorer is compatible with both Notes Pilot and Edit Pad. This means that the tools work successfully in conjunction with the Internet Explorer browser. The Edit Pad tool is compatible with the Netcaptor browser. Table 3.4 shows that the Netcaptor browser is not compatible with the Notes Pilot tool and thus the results are inconsistent with the Internet Explorer Browser results.

Table 3.4: Compatibility between Browsers and Editing Tools

Editing tools / Tools	Notes Pilot	Edit Pad
Internet Explorer	Compatible	Compatible
Netcaptor	Not Compatible	Compatible

3.4.4 Comparison between editing tools

The criteria used to evaluate the editing tools are shown in Figure 3.14. Table 3.5 highlights the strengths and weaknesses of the editing tool and these will be used in specifying the requirements for the prototype. Table 3.5 shows that both editing tools are poor at re-positioning and formatting web content from an HTML environment to a text-based environment. The Notes Pilot tool is also poor in making calculations and returning to the browser. It is excellent at saving work and retrieving old files. The Edit Pad tool is successful at all other activities except re-positioning and formatting web content. It can be seen that tool support is lacking or current web-based tools support the user poorly in a number of areas. The tools assessed do not give a user the provision for making calculations and making notes, or accepting user input.

Assessment criteria for Editing tools

Poor Support

This category is selected for the tools that do not fully support the tasks such as making calculations, and accepting user input. This may mean that the user can not achieve the task in one single step or the tool is not sufficiently flexible to support the task. In some cases the user takes extra steps and it takes much longer to achieve the task with the support of the tool. The user may also have to manually change the format of the document and its content to complete the task. The tools that demonstrated these weaknesses were placed in this category.

Good Support

This category is selected for tools that support many steps of the task at hand. The tool may have limited menu options or commands, which restricts the user from completing the entire task. For example Edit Pad tool will not allow images to be pasted. However, the user can make notes in a text format at any place in the document.

Excellent Support

This category is selected for tools that not only complete the entire task but in addition, allow the user a flexible environment or different menu options to complete the task. An example is the Notes Pilot editing tool which is excellent at retrieving (opening) files and saving files.

Figure 3.14 Assessment criteria for Editing tools

Table 3.5: Comparison of two web-editing tools, Notes Pilot and Edit Pad

Tools / Tasks	Notes Pilot			Edit Pad		
	Excellent	Good	Poor	Excellent	Good	Poor
Reposition and format			*			*
Make calculations			*		*	
Make notes		*			*	
Save all work	*			*		
Print web content		*		*		
Retrieve files	*			*		
Return to browser			*		*	

3.5 Concluding remarks

Current web browser functionality and components have been identified. Users view web browser material on the Web through windows. Users need to compare the selected web sites and decide with the assistance of tools its relevance to the activity. After users have collected relevant information from web sites they will use browsers and window management tools and supporting software to compare the relevant information from multiple web pages. This content is then edited and formatted using editing and word processor tools and then filtered with the assistance of plug-in software; this brings the information together into one common work space. The final outcome is a single web site that has been selected and will be processed further using a browser. A number of web analysis support tools have been identified. The Notes Pilot, Edit Pad and Netcaptor and IE Browser were compared and the shortcomings of these tools were identified. It must be noted that decision-making skills and user perception complement the use of tools. Shortcomings of the tools reviewed will be identified and used in defining the functional requirements of the prototype in Chapter 4.

Chapter 4: Specification and Design of Prototype

This chapter describes the process used to develop the proposed tool. It begins with the requirement specification for the prototype, outlines the design issues and then includes provisional sketches of the user interface of Networker.

4.1 Functional Specification

Chapter 3 reviewed the support provided by current tools for the decision making cycle and studied approaches to the way users process information on the Web and tools available to assist Web users. Features of the tools were rated excellent, good and poor according to the specified criteria. The proposed tool will retain the features in the excellent category, will enhance the features in the good category and will make provisions for the features in the poor category. The literature reviewed in Chapter 2 also provides a basis for the functionality specified. These factors are all considered during the development of the requirements of the prototype.

4.1.1 Required Features of the Tool

The tool must:

- permit the user to access and then keep references to several web sites;
- permit the user to view content from more than one web site at any time;
- provide the user with a working area (a 'WorkPad') onto which web content can be pasted, URLs can be referenced, notes can be made, and calculations carried out;
- permit the user to switch focus between web sites, and between web sites and the WorkPad;

- provide for user control of the number of windows to be displayed at any time, with a minimum of one (either a web site or the WorkPad), and a maximum of three (two web sites plus the WorkPad);
- permit the user to switch between vertical and horizontal arrangement of windows, to suit the information being browsed;
- provide for system placement of windows, with window size(s) being a function of the number of windows being displayed at any time, and the chosen orientation (horizontal or vertical).

Figure 4.1 and 4.2 demonstrate the current and proposed tool support for the web framework. Currently, users collect web content via a web browser such as Internet Explorer and then arrange the browser windows using a Window Management tool. The critical content is copied into an editing tool such as Notes Pilot or Edit Pad. Further analysis is conducted and one web site is filtered out for further processing. Finally, users open the selected web site via the browser. Figure 4.1 illustrates the interaction between the three components and how and where the stages are supported.

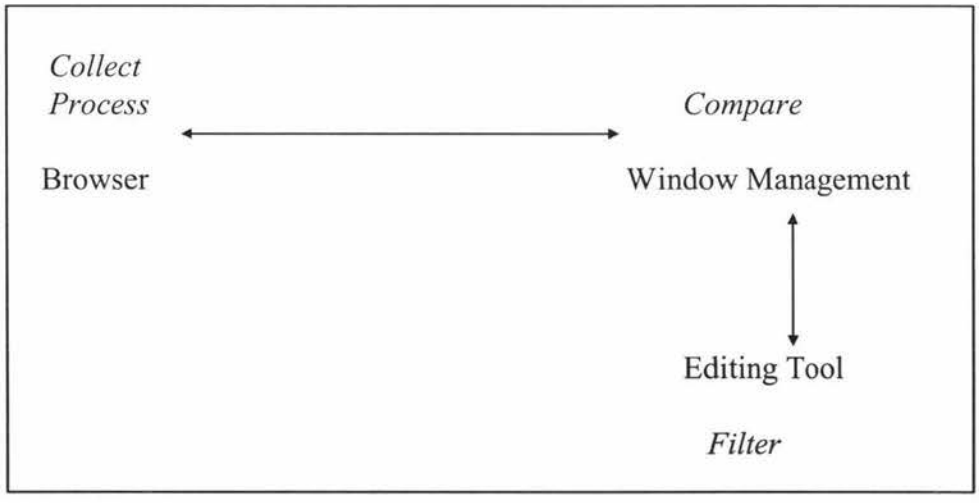

Figure 4.1 Current Tool Support for the Decision-Making Framework

Figure 4.2 illustrates the interaction between two components and how and where the stages are supported. The proposed tool will be integrated with the web browser.

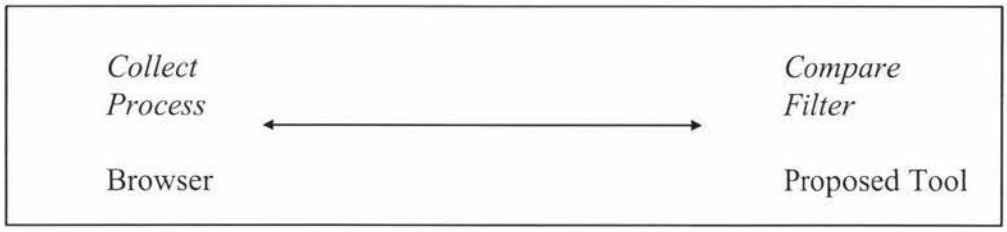

Figure 4.2 Proposed Tool Support for the Decision-Making Framework

4.2 Non-Functional Specification

Dix, et al. (1998) and Nielsen (1993) have developed various usability principles. Five of these principles have been incorporated into the prototype requirements: Learnability, Efficiency, Flexibility, Robustness and Feedback. Learnability defines the ease with which new users can interact with a new system and achieve maximum performance. This category is decomposed into the sub-categories of familiarity, consistency and predictability (Phillips et al, 1998). Efficiency is a measure of the productivity of a system. It measures the number of actions needed between the user and system to complete the task. This may be measured in time or number of operations. The principle can be decomposed into the sub-categories of simplicity, responsiveness and task conformance. Flexibility relates to the numerous ways a user and system can exchange information. It includes the sub-categories of dialog initiative, multi-threading and customisability. Robustness is the quality of support the user is given in achieving a goal. It includes the sub-categories of error-prevention, recoverability and provision of help. Feedback is the final principle, which measures the quality of the response provided by the tool to the user. It includes the sub-categories of observability, and the quality of both the presentation and content of the information displayed by the tool.

4.2.1 Required Non-Functional Features of the Tool

The tool must:

- be consistent with the user interface of the browser (learnability);
- provide for each of the following functions to be carried out by the user in a single operation (efficiency):
    - opening browser windows;
    - opening web sites;
    - switching focus between displayed web sites;
    - switching focus between displayed web sites and the WorkPad;
    - revisiting previously accessed web sites;
    - changing the number of windows displayed;
    - switching between vertical and horizontal arrangement of windows;
    - transferring web content to the WorkPad;
- make the system status clear to the user at all times (feedback);
- provide user help and support (robustness).

4.3 Design Issues

This section will explore a number of alternatives in the design of the prototype. The different approaches are evaluated and the pertinent requirements are implemented in the final prototype.

Lo-fidelity (lo-fi) prototyping is a common technique used in the early stages of the design process (Rettig, 1994). It is used to build a mock up of the interface that may be tested by users. The process may involve hand-drawn paper prototypes or computer

produced drawings that may show a flow from one screen to the next. It is possible to 'walk through' a paper prototype using a scenario or use case, and uncover strengths and weaknesses in the design. Software can be built in an iterative fashion through the development of prototypes. An initial lo-fi prototype can be used to refine the requirements with the users, and then discarded prior to the development of the system.

A number of factors were considered during the design of the final tool interface. These include placement of widgets such as browser windows, tabs and URL boxes.

One browser window vs. Two browser windows

The option for users to have just one browser window open means that the web site can be easily read with limited vertical scrolling involved. It will also be easier to find the content needed to transfer onto the work area. The option of presenting two web sites at a time enables the user to compare content before transferring content onto the work area. Both horizontal and vertical scrolling will be required. This can be tedious for the user.

Decision: In the final prototype the user will be given the option of having one or two browser windows displayed at a time. A maximum of two windows will be displayed at any one time. More than two windows will reduce the quality of the content and hinder any form of comparison. The option of allowing multiple windows at a time will lead to increased user flexibility and provide an effective way to compare web content from multiple web sites.

Tabs vs. Buttons vs. Url Boxes

The use of tabs to present previously visited websites is very useful. Users are generally familiar with the tab structure used to organise multiple menus in current web sites. It is a form by which users can access hidden windows in one step. This mechanism is similar to that used in Netcaptor, a commercial browser available on the Internet. Buttons require more room and space in between to distinguish between

them. A limited number of buttons can be arranged below a browser. The active button may prove to be difficult to interpret.

Tabs can be in iconised form representing hidden windows as in Microsoft Office Applications, or tabbed windows showing the web site names in the labels and displaying the current window with the other windows hidden, as in Netcaptor's interface. URL boxes are used to select previously visited URLs so users do not need to recall the URL names when accessing the same websites more than once. The user can select and click in one step. This is also how conventional web browsers such as Internet Explorer and Netscape Navigator work. These browsers also give users the option to manually type in the URL in the URL box.

Decision: Both tabs and URL boxes will be used in an attempt to maintain flexibility. Users can either select or type URLs to access visited web sites or click on tabs to access the hidden window.

Vertical vs. Horizontal Window Arrangement

It is common to use a vertical window arrangement for documents created in a portrait layout. Users read from left to right and top to bottom and therefore, a vertical window arrangement is mainly used. Microsoft applications use both horizontal and vertical window arrangement, although users most often use vertical scrolling to move from one page to the next. Horizontal scrolling may be used for reading web content in landscape layout.

Decision: Each website has a different layout and so given the choice of horizontal or vertical arrangement users can switch between the layout they prefer in one step. A user may create an arbitrary number of browsers that can open multiple web sites, however only two browsers can be viewed at a time.

Keep work area open all the time vs. Keep work area open as an option

A restriction will be imposed on the user if the work area is open all the time. The space in the interface will decrease. However, if the work area is available to the users all the time then it is one less option to remember to use. If the work area can be closed this allows the user to control the placement of the work area and thus, more space is available to users to compare web content in the windows.

Decision: Users will have the option to close WordPad or leave it open on its own, according to their preference. The system will resize the remaining windows accordingly.

4.4 Snapshot of the Prototype

In light of the requirements and the design rationale described in previous sections a sketch of the interface for Networker is developed.

The interface can display from one window (See Figure 4.3) to a maximum of three windows. Networker has controls, which can be selected to manipulate the browser window(s). Moving from left to right, the first two controls (horizontal and vertical arrows) allow the user to switch between horizontal and vertical layouts. The next control allows the user to add browser windows, a maximum of two windows at any time. The final control allows the user to hide the WorkPad or show the WorkPad if it is hidden.

When the user has two windows and a WorkPad open in vertical format, he/she can easily transfer web content across to the WorkPad and edit the content (See Figure 4.4). In contrast, Figure 4.5 shows two windows and a WorkPad open in horizontal format. Figure 4.6 shows two web sites in a horizontal layout with the WorkPad area hidden.

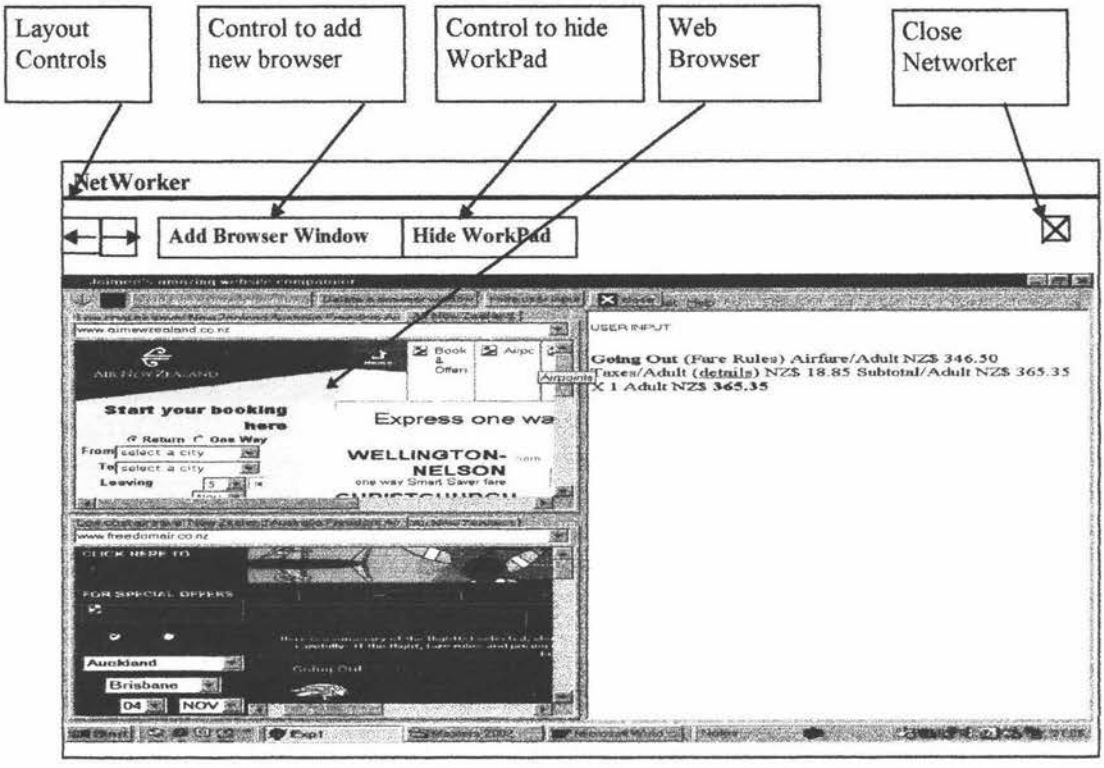

Figure 4.3 One open website

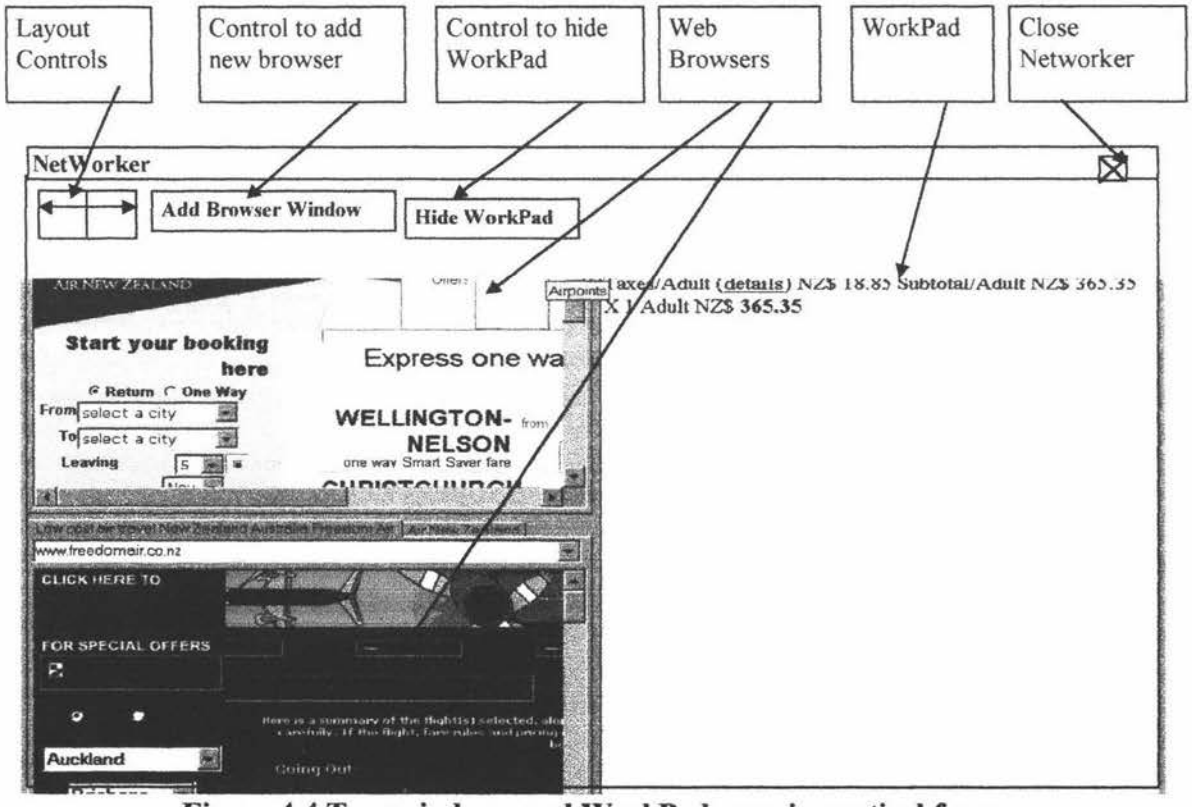

Figure 4.4 Two windows and WorkPad area in vertical form

Figure 4.5 Two websites and a WorkPad area in horizontal form

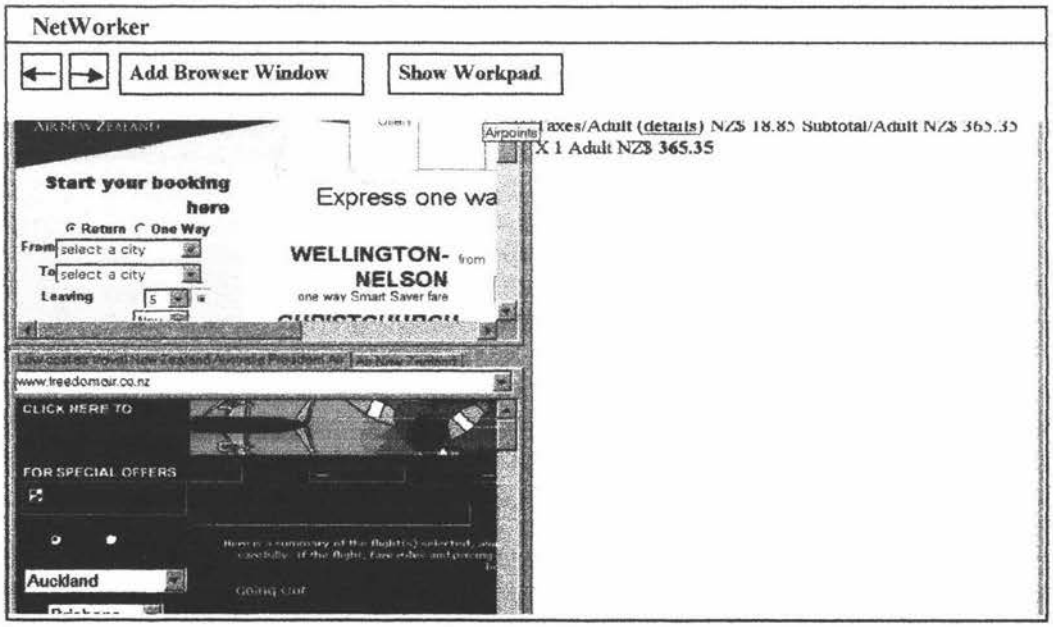

Figure 4.6 Two websites in horizontal form with hidden WorkPad

4.5 Concluding Remarks

The functional and non-functional specifications have been identified. A number of design issues have been discussed and design decisions made. Sketches have been produced showing a provisional design for the user interface of the Networker. The next chapter will describe the approach used in implementing the prototype.

Chapter 5: Implementation of Prototype

This chapter describes the implementation of the interface designed in Chapter 4. Many design decisions were made and a review of the non-functional and functional requirements undertaken. The chapter ends with an example scenario which shows the tool in action.

5.1 Environment

The program called NetWorker was written in Borland's Object Pascal using the VCL (Visual Components Library) and implemented using version 7 of Borland's Delphi integrated development environment (IDE) (Jacobson, 1999). This is a Graphical User Interface (GUI) environment. It includes provision for creating and manipulating web browsers. Delphi was selected as the programming environment as it is a visual programming environment that uses drop down widgets to develop applications and uses skeleton templates. This is ideal for building quick and easy lo-fi prototypes in a restricted time frame. A number of programming obstacles were faced during the development of the prototype. Some of these issues were resolved whereas others could not be due to time constraints (See Appendix A for program code).

5.2 Implementation Issues

There were a number of implementation issues to be considered so that the principal functional and non-functional requirements could be fulfilled. Internet Explorer is the browser that is used to access the web sites.

Manipulating Web Browser Windows

The browser window incorporated into the tool is not the standard system-created window. Due to the fact that the system generated components have limited procedures and properties, some of the components such as web browsers, and URL boxes are user generated locally in Delphi, using Tcontrol function and its inherited

parent properties. In order to manipulate these windows, parent properties must be called and global variables used.

Initially, the approach was to hardcode two browser windows that could be displayed or not according to the user's selection. Delphi has its own standard web component that allows for this approach. It became obvious that the only way to get a functional system was to create dynamic browsers and allow for each browser to have its own controls so that the user could add, close, layout and control the dimensions of the browser window. In this program, very little skeleton code was used and so the prototype took a lot longer to code than anticipated.

Networker needs to know where to place a new browser window depending on the number of panels already present in the screen. Networker also needs to know whether the screen is in a vertical or horizontal layout. According to the feedback, Networker assesses the dimensions the browser window should be (either half or full), the position to display it and the layout orientation of the browser window.

Acquiring Last Page Visited

Networker is a front-end application to Internet Explorer. Internet Explorer has a constraint that it saves the URL the user visits instead of the last visited page. Therefore, when the user wants to access the last web page visited in Networker, the homepage of the website re-loads in the browser window. As a consequence, the URL box saves the URL typed in by the user and does not add on extended links of each web page visited in a site. This is a flaw in the prototype that can be fixed given time and it must be functional in the final tool.

Accepting User Input in WorkPad

The user input section is Word compatible. HTML content can be directly transferred into this area. The generic title of the web sites is used to name the tabs that represent each web site minimised or restored on the window. WordPad is used instead of Word as Word opens in the internal directory and not directly in the Delphi design area. A text editor, with WordPad functionality is displayed beside the window section. This allows users to drag and format web content into one workspace. The toolbar in the

WorkPad is not yet functional but demonstrates the options that can be made available in the final tool.

Minimise and Restore Control in the WorkPad

The minimise and restore control in the WorkPad is bi-modal. When the WorkPad is showing the minimise WorkPad label appears and vice versa. However, there is a lot of coding that lies behind this control. The 'minimise' control displays a rectangular shaped icon in the top right side of Networker. The 'restore' control has a lot of technicalities associated with it. Networker needs to know where to place the WorkPad depending on the number of panels already present in the screen. Networker also needs to know whether the screen is in a vertical or horizontal layout. According to the feedback, Networker assesses the size WorkPad should be (either half or full), the place to display it and the layout of the WorkPad.

Opening Multiple Websites in the Browser Window

Conventionally, users are capable of opening multiple websites in a browser window and therefore this had to be supported. A number of new components had to be created and inherit the properties of the Web Browser window in Delphi. The browser window had to have a URL box inside it so that the user can type in URLs, causing websites to be opened and displayed. The older websites then had to be minimised and a tab component was created to achieve this. The tabs had to be labelled with the Web site Location Name. Once the user enters a URL the server shows the links it traverses to reach the website. These links are shown while the server is searching and then once the whole web site is displayed the web site Location name is shown in the tab label.

Maintaining a URL list

The URL box is a component created in association with the Web browser window. Each window has one URL box. The URL box saves all the URLs the user enters and allows the user to select from a list of URLs to access those websites.

Closing Browser Windows

Initially the user had to use a bi-modal button, which added or deleted browser windows. Up to two windows were displayed at any point in time and thus, when the user selected the 'delete browser window' control, the lower browser window would be deleted. A major change was made, such that each browser window has a close button on the left side of it, which removes the browser window the user selects. This approach was very time-consuming to code since the position of each close button had to be consistent in all the browser windows and the control had to be coded as a component of the browser window.

Login Issues

As Networker is a front-end to Internet Explorer the user is required to login during the connection into the Internet phase and again for Internet Explorer. This is tedious, but has not been changed due to time constraints in developing the prototype.

General Layout and Appearance

The interface is grouped in two sections. The right side displays the WorkPad and its controls and the left side displays the websites and its controls. The background colour is pale green and the website browsers and WorkPad have a silver colour. This colour scheme is consistent with Microsoft products. Users are familiar with a similar combination in all Microsoft applications. Nielsen's (1997) website uses pale colours and uses different colours to differentiate the main sections if the website.

5.3 Requirements Review

Looking back at the functional requirements specified in Chapter 4, Table 5.1 shows that most of the requirements have been implemented.

Table 5.1 Review of Functional Requirements

Functional Requirement	(Yes/No)
permit the user to access and then keep references to several web sites	Yes
permit users to view content from more than one web site at any time	Yes
provide the user with a working area (a 'WorkPad') onto which web content can be pasted, notes can be made	Yes
permit the user to switch focus between web sites, and between web sites and the WorkPad	Yes
provide for user control of the number of windows to be displayed at any time, with a minimum of one (either a web site or the WorkPad), and a maximum of three (two web sites plus the WorkPad)	Yes
provide the user with a working area (WorkPad) onto which URLs can be referenced, and calculations carried out	No
permit the user to switch between vertical and horizontal arrangement of windows, to suit the information being browsed	Yes
provide for system placement of windows, with window size(s) being a function of the number of windows being displayed at any time, and the chosen orientation (horizontal or vertical).	Yes

The functional requirements concerning the customised controls of *Networker,* such as the down and right arrow; the addition of browser windows control and the WorkPad minimise and restore control have been implemented. However, the generic controls on the toolbar and the open, close, save and print controls have not been implemented.

Looking back at the non-functional requirements mentioned in Chapter 4, Table 5.2 below shows that most have been implemented.

Table 5.2 Review of Non-Functional Requirements

Non Functional Requirement	(Yes/No)
be consistent with the user interface of the browser	Yes
opening browser windows in a single operation	Yes
opening web sites in a single operation	Yes
switching focus between displayed web sites in a single operation	Yes
switching focus between displayed web sites and the WorkPad	Yes
revisiting previously accessed web sites	Yes
changing the number of windows being displayed	Yes
switching between vertical and horizontal arrangement of windows	Yes
transferring web content to the WorkPad	Yes
make the system status clear to the user at all times	Yes
provide user help and support	No

5.4 Screen Snapshots

This section explores the prototype that has been created from the list of requirements in Chapter 4. Section 5.3 described all the requirements that have been and have not been implemented. Some limitations were found while using the Delphi environment and these have been recorded in Section 5.2. A brief review of the components of Networker and the different panes is illustrated below. The tool's functionality may be seen in light of the decision-making framework created in Chapter 2.

Figure 5.1 shows Networker with one browser window open. There is one website open in the pane, Freedom Air, and another website is minimised, Air New Zealand. Users can enter a URL or select one previously visited via a URL box. Users may also click on the tab above to access minimised websites. Each browser window has a close button on the top right corner. The browser window is open in a vertical layout

as the down pointing arrow in the first control is inactive. A user may add a new browser window by selecting the "Add browser window" control. Currently, the WorkPad is minimised but its toolbar can be seen above the browser window. There is a control available to show the WorkPad. The generic controls on the WorkPad toolbar are not implemented. Finally, on the right side of the tool there is a control labelled "Close" to close Networker.

Figure 5.1 Networker with one browser window display

In the **Collect and Compare Phase**, the user has collected the relevant websites and is looking at the websites, either on a vertical layout or horizontal layout. In Figure 5.2 two web browser windows are displayed vertically and the WorkPad may be used to copy relevant content for comparison . The user has collected two web sites ready for comparison.

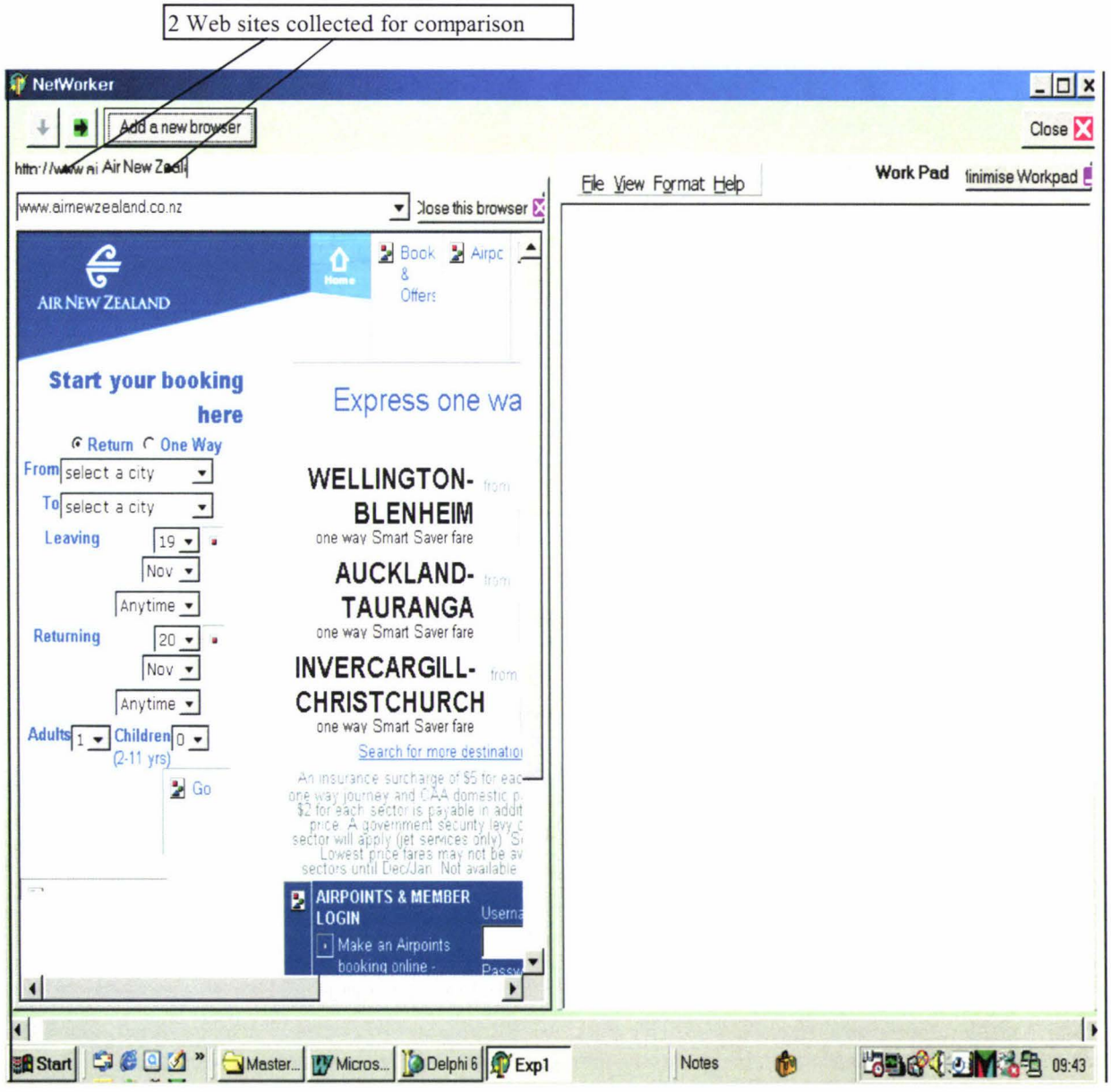

Figure 5.2 Supporting the Collect and Compare Phase

Figure 5.3 shows the **Filter Phase**. It is a display of one website and the WorkPad in Networker. The down pointing arrow control is inactive and will remain this way until the user selects the horizontal control. The user has copied text from the browser window showing the Qantas website into the WorkPad area.

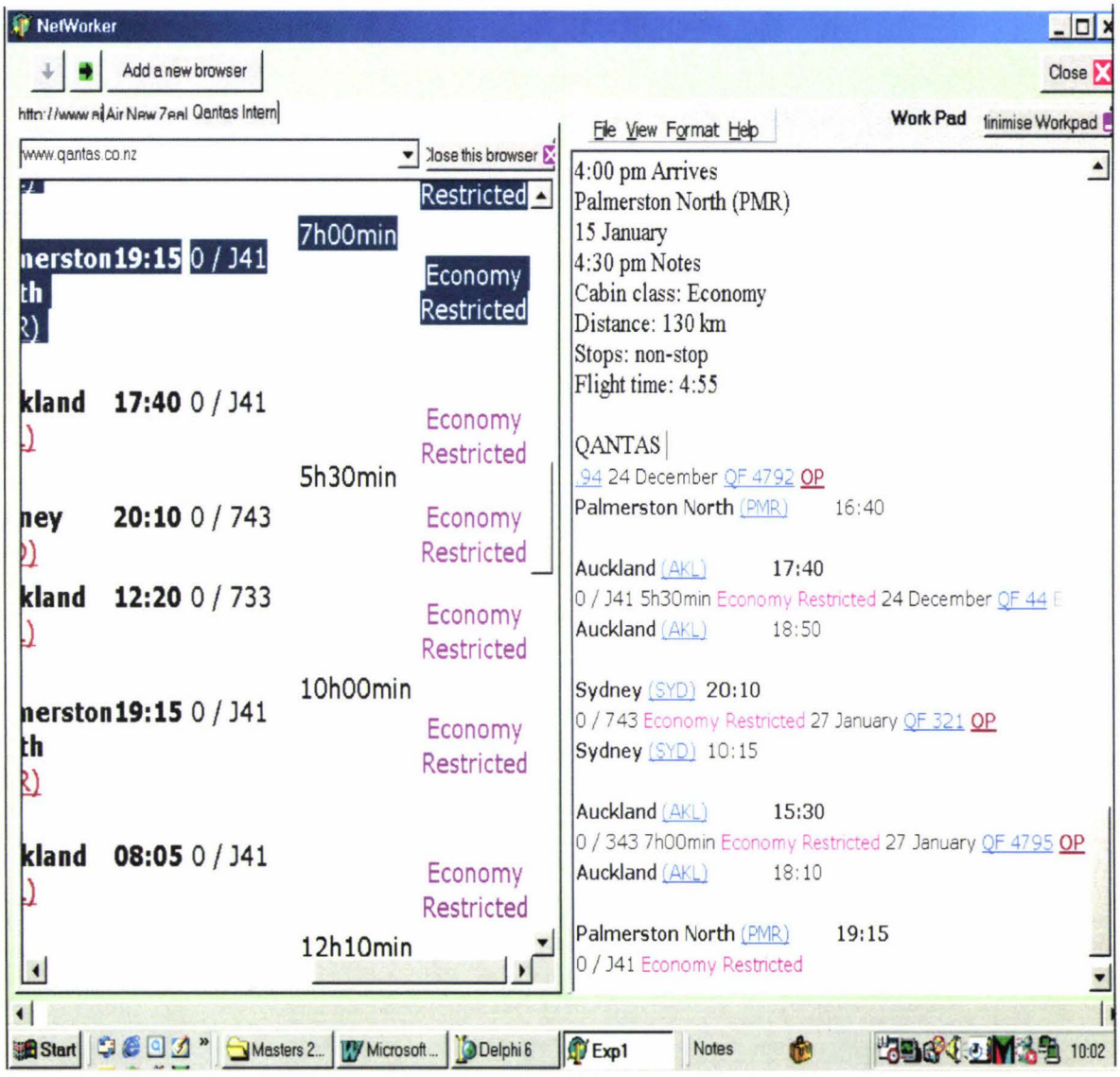

Figure 5.3 Supporting the Filter Phase

Finally, the **Process Phase** is where the user analyses the content in the WorkPad and then returns to the browser window to purchase an airline ticket. Figure 5.4 shows a full view of the Freedom Air website.

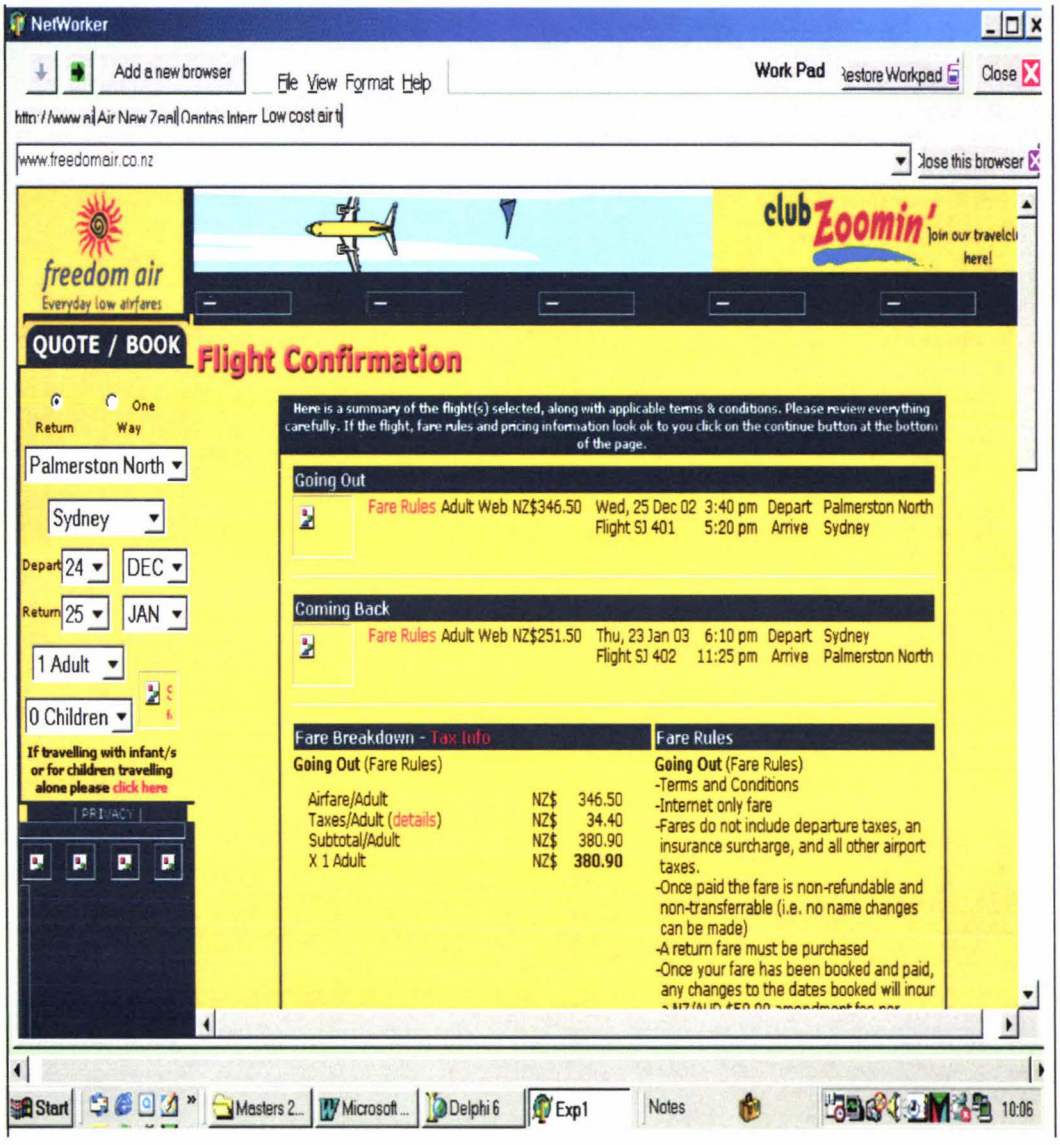

Figure 5.4 Supporting the Process Phase

5.5 Concluding remarks

Delphi has been used as the development environment to implement the prototype. All the implementation issues and limitations of the prototype have been discussed in this chapter. The requirements that have been implemented have been listed. Finally, screenshots of the final prototype have been presented using a scenario to illustrate the main features of Networker.

Chapter 6: Evaluation of Prototype

The first section of this chapter describes user evaluation techniques and emphasises interviews and user observations as techniques to obtain feedback on the prototype. The second section of this chapter describes the methodology and instruments applied in this research. The third section covers the results and analysis of the user evaluation. From this analysis, minor changes were made to the tool, which is described in the last section of this chapter.

6.1 Tool Evaluation Techniques

Interface evaluation can be undertaken for many reasons and there are a number of techniques available (expert reviews of several kinds, observation, questionnaires etc.) (Shneiderman,1998a). For this reason it is important to identify the high-level goals to guide the evaluation (Preece et al., 2002). The purpose of the evaluation is to determine the strengths and weaknesses of the prototype. It is useful in these circumstances to observe users whilst they are carrying out typical tasks and to time their actions (Preece et al., 1994).

An interview is a structured and direct means for gathering information (Dix and Finlay, 1993). The advantages of interviewing include the fact that the evaluator can vary the level of depth suitable to each area and if necessary some questions can be taken to more depth if the findings are useful. Interviews typically use the top-down approach, whereby general questions are asked in the beginning and progress to more detailed questions in the end. Interviews are effective for high-level evaluation where analysts are looking for open-ended feedback and need elaboration in areas like user preference, impressions and attitudes. The interview may highlight or clarify problems with the system that were not shown in other evaluation techniques. To be effective, an interview must be planned with a set of consistent questions prepared.

User Testing enables developers to examine whether users can complete the main tasks using software (Preece et al, 2002). Developers can observe how ordinary users use their tool and identify the routes taken and the strengths and weaknesses of the design according to the user. Testers carry out *usability testing*, in which the interface is studied under real-world or controlled conditions, with evaluators gathering data on problems that arise during its use. The actions and feedback of the interface are compared to the user's goals and knowledge, and differences between the user's expectations and the steps performed by the interface are noted.

Many practical issues have to be thought about in advance. The location of the tests need to be prepared, the equipment needs to be tested and an informed consent form should be available for users to read at the beginning of the study (Preece et al, 2002). The tester needs to state the initial test goals and questions clearly, select a framework to guide an activity in the field, decide how to record events, and analyse the notes. This usability observation process can consist of a think aloud protocol. This is where users are asked to elaborate their actions by 'thinking aloud': describing what they believe is happening, why they take an action, what they are trying to do (Dix and Finlay, 1993). The advantages of using this technique include the fact that an amateur can conduct this session and gain very useful findings on the user's interaction with the system. This technique can be used for evaluation throughout the design process using paper or simulated mock-ups for the earlier stages. The result may provide a subjective view from the perspective of a sample of user opinions. The user's actions may be recorded using paper and pencil and audio.

Questionnaires are an evaluation technique used to gain better understanding of user opinions. They are familiar and inexpensive (Shneiderman, 1998a). Both closed and open questions can be asked. However, in longer questionnaires, open and specific questions follow on from closed and general questions. Questions can also be grouped into sub-topics, so that the questionnaire is more easily read and can be completed in a logical manner. The biggest advantage is that a usability questionnaire gives you feedback from the point of view of the user (Kirakowski, 1996). Questionnaires can be distributed to a large sample of people and the statistics can be fruitful for research. Additional advantages are that questionnaires are usually quick and therefore cost effective to administer and to score and that you can gather a lot of

data using questionnaires as surveys. The biggest disadvantage is that questionnaires may not always be reliable as the results tell you only the user's perception of the situation.

There are a number of styles of questions that can be included in the questionnaire (Dix and Finlay, 1993):

General

These questions establish the background of the user. Data may be collected on age, gender, qualification, occupation, and experience with computers. Questions can be designed in an open-ended, scalar or multi-choice manner.

Open-ended

These questions ask for the user's subjective opinion in an area. The user will write a couple of sentences on any aspect of the system asked in the question. These responses can be useful for gathering subjective information but not for critical analysis. The responses may identify future improvements for a system.

Scalar

The user is asked to judge a specific question on a numeric scale. The scale will pertain to the agreement or disagreement with the statement. The granularity of the scale may vary e.g. from 1 to 5. This may provide varying levels of agreement but it is difficult for the analyst to interpret in a qualitative manner. An example is the Likert scale (Kirakowski, 1996).

"Likert scaling is quite tricky to get right, but when you do have it right, you are able to sum the scores on the individual items to yield a questionnaire score that you can interpret as differentiating between shades of opinion from *'completely against'* to *'completely for'* the construct you are measuring." (Kirakowski, 1996, Section 14)

It is possible to find questionnaires, which seem to display Likert-style properties in which many of the items are simply re-wordings of other items.

Multi-choice

The user is offered a specific number of responses to a statement and is asked to select one or more of the answers as appropriate. The user indicates the answer by writing a letter or ticking or filling in an empty box or circle.

Ranked

These questions are designed to get the respondent to place an order on items in a list.

6.2 Methodology

The goal of this project is to evaluate the interface of Networker. The results of the evaluation will be summarised and recorded. The users will be asked to use Networker to search for the best solution to common tasks Web users conduct on a regular basis. Users will be asked to "think aloud" as they use Networker. Users will be observed and will walk-through a number of scenarios using the tool. The users will be timed during this exercise and the tester will record the performance. It is hoped that all the functionality of the tool will be tested during the tests. After the completion of the tests users will be requested to complete a questionnaire regarding their experience with the tool. The purpose of this evaluation is to find the strengths and weaknesses of the prototype. The common findings will be prioritized and incorporated in the final prototype. A number of common findings will be used for developing areas for the tool in the future.

First, the author explored the common tasks a typical user may undertake on the Internet. After much consideration the author selected three broad tasks and designed a recording matrix by which the actions of the users can be recorded. The author used the matrix to record the time to complete the task, whether the task was completed within the time frame, i.e. a success, or incomplete within the timeframe, i.e. a failure. The matrix allowed the author to record the number of times Browser 1 and 2, the horizontal view, vertical view, WorkPad, and visited websites were accessed (See Figure 6.1). A pilot experiment using the set tasks and the matrix allowed the author to refine both the procedure and tasks prior to the actual experiment.

Task #	Time (min)	Success / Failure	Browser 1	Browser 2	Horizontal layout	Vertical layout	Total WorkPad accesses	Visited websites accessed

Figure 6.1: Results table

A tape recorder was used to record the verbal feedback the users made while conducting the tasks. The users were told to "think aloud" and inform the author of their thoughts at any point in time. The author using the tape recorder recorded these and transcriptions were developed using pen and paper. The users were also asked to give their written feedback in a questionnaire where their feedback was related directly to the "think aloud" comments.

Three Computer Science postgraduate students and two Computer Science staff members who all had experience with the web were asked to evaluate Networker. They were representative of people who regularly use the web to carry out tasks. Informed consent was obtained from the users who had all the evaluation procedures fully explained in advance. They were also told that it was the system that was evaluated and not their actions. Documentation about Networker was made available to the users who were allowed to explore the system for a quarter of an hour before the evaluation properly began (See Appendix B for more detail).

There were three tasks that users were asked to complete in a set time frame. These tasks are common activities conducted by web users.

TASK A

Find the most competitive price for The Lord of the Rings (Leatherette Collector's Edition), in NZ dollars, including freight costs. Jane needs the book before Christmas! Check acceptance of international credit cards.

TASK B

Find the most competitive home loan interest rate and period New Zealand banks provide for a customer wishing to purchase a home.

TASK C

Find a single room in a Christchurch motel that has a good reputation and offers standard services at the best price. The price range should be between NZ $70-$90.

Users were told that they had 10 minutes to complete a task. Users were observed and asked to "think aloud" about the process as they used NetWorker with their comments recorded. The author observed their actions and made notes. Whilst the tasks were being completed, users were timed and the observer noted how many times the WorkPad was accessed, how frequently horizontal and vertical layouts were selected and whether the "add new browser" button was selected.

At the end of the session, users filled out a questionnaire regarding their experience with the tool. A Likert scale from 1 to 5 was used where 1 represented *strongly disagree* and 5 *strongly agree*. There were 13 items in the questionnaire (See Table 6.1). Some of the items in the questionnaire (2,7,9,10,11) were based on those in SUMI (Porteous et al., 1993), which considers issues such as efficiency, control and affect, but others had to be added that specifically related to the Collect and Compare activities (3,4,5 and 6). Finally, the value of the tool (1 and 12) as well as its novelty (13) was ascertained.

Table 6.1: Questionnaire on Networker

No	Item	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	It is clear what the purpose of the tool is					
2	It is clear how to use the tool					
3	It is easy to collect information using the tool					
4	It is easy to compare multiple web sites using the tool					
5	It is easy to refer back to visited web sites					
6	It is easy to copy important information and edit using the WorkPad					
7	It is easy to make the software do exactly what you want					
8	The work space of the tool can be customised to my needs					
9	The labels and icons on the buttons can be clearly read and are understandable					
10	The tool has an attractive presentation					
11	The software responds too slowly to inputs					
12	The tool will be useful to my academic or personal activities on the Net					
13	The tool is very similar to other tools I have used					

The questionnaire also had a set of open-ended questions. The questions asked for recommendations on any changes that could be made in the overall design, language, and controls of the tool. The users were asked about their perspective on their performance on the tasks and to record their opinion on the best and worst aspects of the tool.

6.3 Results

It can be seen from Table 6.2 that the mean time for carrying out the first task was longer than that for the other tasks and that two people failed to complete it. The mean time for the other tasks was substantially lower. The WorkPad was used regularly, 28 accesses overall with a mean time of 5.6 times. Only three of the users selected a second browser window, the others preferring to work using one window only. Information from a site would be copied to the WorkPad and then another site selected in the same window. All but one person selected both the vertical and horizontal layout at some time in the proceedings. The other person always used the vertical layout.

Whilst the sample is very small, some trends emerged (See Table 6.3). The results of the questionnaire showed that the median for item 12 was five, indicating that most of the users strongly agreed with the proposition that the tool would be useful for personal or academic activities. The users disagreed with the statement that the tool was similar to other tools used, recognizing its novelty. The medians for each of items 1, 2, 3, 4, 6, 8 and 9 were four, indicating an overall level of satisfaction. The users were less satisfied about *referring back to websites, getting the software do exactly what they wanted* and *arranging the work space to meet their needs* (all with a median of three). All users disagreed that the software responded too slowly (See Appendix C for more detailed results).

Table 6.2: Evaluation Results

Task #	Mean time (mins)	Number succeeded	Total WorkPad Accesses
1	8.96	3	8
2	6.14	5	10
3	7.2	4	10

Table 6.3: Analysis of Results

Item Number	Item	Median
1	It is clear what the purpose of the tool is	4
2	It is clear how to use the tool	4
3	It is easy to collect information using the tool	4
4	It is easy to compare multiple web sites using the tool	4
5	It is easy to refer back to visited web sites	3
6	It is easy to copy important information and edit using the WorkPad	4
7	It is easy to make the software do exactly what you want	3
8	The work space of the tool can be customised to my needs	4
9	The labels and icons on the buttons can be clearly read and are understandable	4
10	The tool has an attractive presentation	3
11	The software responds too slowly to inputs	2
12	The tool will be useful to my academic or personal activities on the Net	5
13	The tool is very similar to other tools I have used	2

The common problems that emerged from an analysis of the transcripts are as follows. All of the participants wanted to be able to directly resize windows (browser, WorkPad). Four people mentioned the lack of a Back button. Two amplified this problem, noting that the interface lacked typical browser features. Three disliked the fact that the last level visited in a website could not be accessed. Two people

complained that they had to log in to both Networker and Internet Explorer. Two people noted that the Close button was duplicated. When asked to comment on aspects of the tool that they really liked, four users mentioned the WorkPad facility and three users mentioned the ability to return to previously visited websites. Finally, two of the five users appreciated the option of using two browser windows.

Users generally showed a liking for the Microsoft Windows overlapping windows arrangement rather than Networker's tiled window arrangement. In a tiled format users cannot resize the windows and many were trying to do so. Users are more accustomed to an Internet Explorer browser and expected to see common browser controls used in the tool. e.g URL names, bookmarks, history, Back and Forward buttons. It was apparent to a few users that there is also no status bar present as in a typical web browser and thus, a few users became frustrated that they could not see the browser status. Some users stated a preference for the hide and show window functionality rather than minimise and restore controls on the WorkPad. Some websites use pop-up windows to display the results of their search, this is not provided by Networker. These windows are opened in Internet Explorer and displayed at any random place on the Networker screen. Most of the users requested more colour and fancy fonts for the tool. A few users wanted to copy graphics as well as text. The labels and icons on the controls were not intuitive and so users were looking for hints on each control.

In reference to the open-ended questions in the questionnaire, there were a number of interesting findings. One of the participants felt that the minimised WorkPad window could be a smaller size similar to Microsoft's minimized window size, the double log-in was tedious, the windows could be overlapped and resized instead of the fixed tiled layout, the system status and speed was not visible, and that the WorkPad facility was useful but graphics could also be incorporated. The concept of Networker is unique and the participant had never come across any similar tool. Additional responses are shown in Table 6.4.

Tables 6.4 and 6.5 present positive and negative responses provided by the participants.

Table 6.4: Positive responses from participants

Participant 1: The option of using two browser windows was the best thing about the tool.
Participant 2: The clean uncluttered interface is the best feature. The concept of the WorkPad is very useful. The web site tabs, which access previous web sites, are very useful.
Participant 3: The option of using two browser windows was the best thing about the tool. The WorkPad functionality is also useful.
Participant 4: Using tabs to look back at previous web sites is wonderful. It can provide a quick comparison even if the content will not be recorded. The name WorkPad is a very good one.
Participant 5: Recording web content on the WorkPad and typing in the source of the information is the best feature of the tool. It is also useful that users can refer back to previously visited websites. Both tabs and combo boxes are a handy mechanism. The option of having split screens, as well as maximising the WorkPad or browser window, is also designed very well.

Table 6.5: Negative responses from participants

Participant 1: There should a search list of web sites that users can go back and visit. The WorkPad and browser window could not be resized.
Participant 2: The WorkPad should be displayed on the interface all the time
Participant 3: The interface is not user friendly. It could be more similar to the Internet Explorer browser, with similar toolbars, presentation, bookmarks and back and forward buttons. The full URL should be shown rather than just the upper hierarchy.
Participant 4: There should be a Back button and windows should be resized. There is no need for a duplicate Close button for the tool. WorkPad could have a different font, the lines could have single spacing and users should be able to change font later if required.
Participant 5: The tool needs more colours with fancy font. The tool does not have Bookmarks or Back and Forward buttons. Each button could have hints or descriptions about its purpose. Users should be able to write in words in the URL box as well as specific URL locations.

6.4 Changes to the Prototype

The feedback from the users led to several suggestions for improvements, some of which have been incorporated into the interface (See Figure 6.2). These are detailed below:

Tool tips

The users who evaluated NetWorker had no prior experience with interfaces that display multiple browser windows, and allow a user to create and delete browser windows; one user did not understand the controls for accomplishing these tasks. Accordingly, all the controls now display tool tips containing brief descriptions of their functions, whenever the cursor hovers over them for 2.5 seconds or more. An example is the tool tip on the vertical layout control button (See Figure 6.3).

Close Buttons

The evaluation version of the software supported two types of Close buttons, one for closing the whole Networker application, and one (replicated in each browser window) for closing an individual browser window. Although the background colors of the application Close button and the browser Close button differed, users confused these buttons. Accordingly, the "Close this browser" button caption has been removed and now shows the close logo. The Close button in the top right corner of the application has also been removed.

WorkPad Button

The evaluation version had a Minimise/Restore toggle button to control the visibility of the WorkPad. The label and the Micorosoft button logo used on the button confused users. This has been changed and is now called the Show WorkPad/Hide WorkPad button.

Font

The font of the headings is larger in size and is in bold. For example the label on the Add Browser Window button and the Minimise WorkPad button have been changed.

Colour

Background color has been changed from the Windows standard beige to a green color for the WorkPad areas and a blue color for the browser window area. These colors were selected from Nielsen's website guidelines (Nielsen, 1997d).

WorkPad Panel

The controls in the WorkPad panel have been grouped. There is a toolbar with File, View, Format and Help on the left side and the Hide/Show WorkPad on the right.

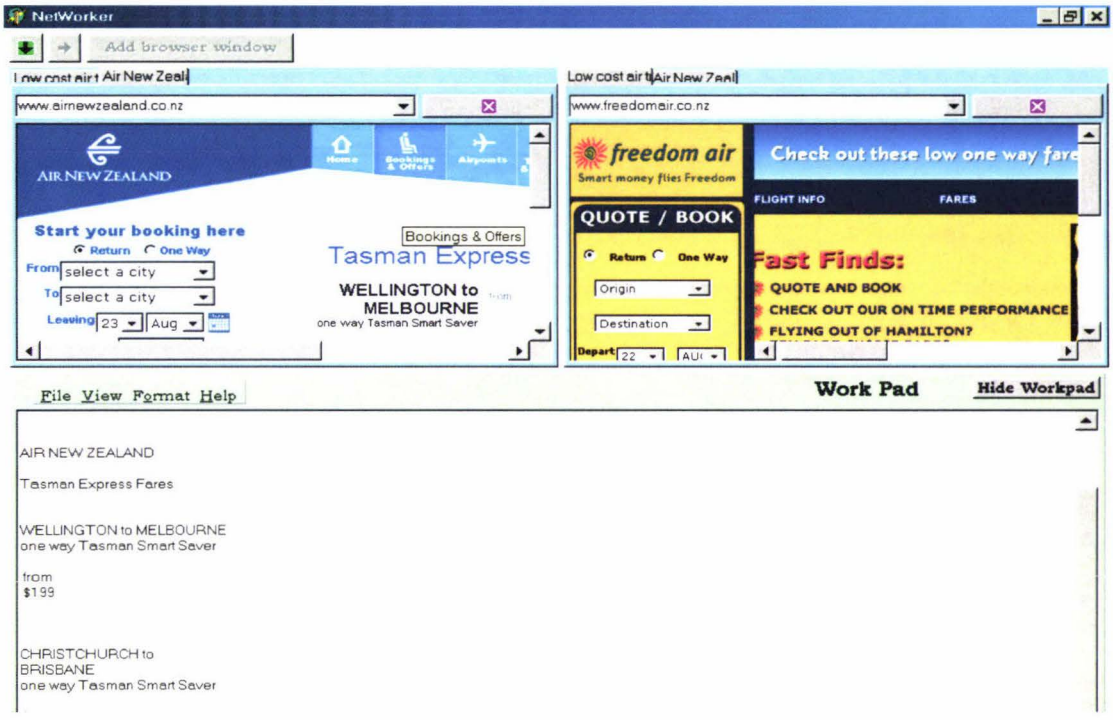

Figure 6.2: The static appearance of Networker with the changes made (2 browser windows and WorkPad)

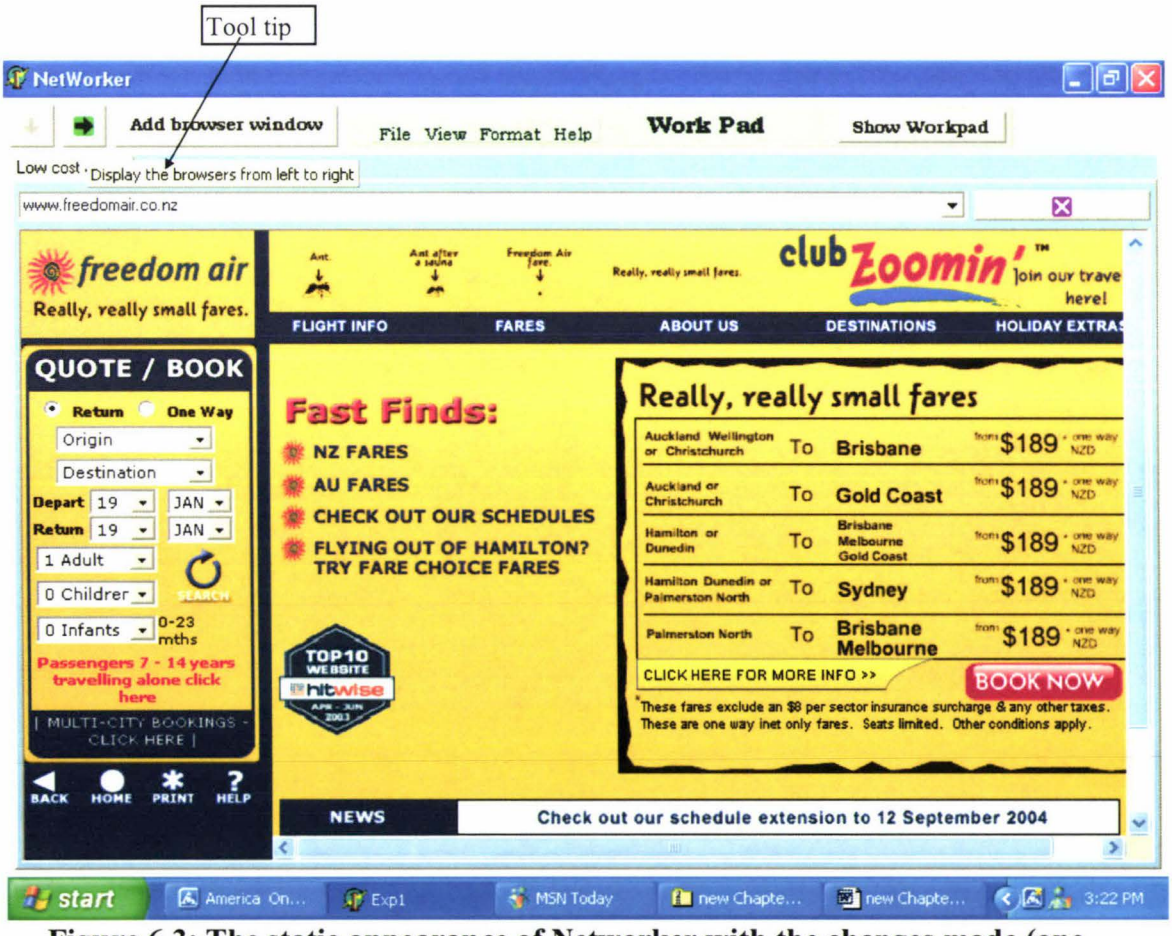

Figure 6.3: The static appearance of Networker with the changes made (one browser window)

6.5 Concluding remarks

User testing, questionnaires and think-aloud evaluation techniques have been used to evaluate the prototype. The methodology used to evaluate the users has been described in detail. The results have been recorded and analyzed. Common problems in Networker have been identified and changes made to the current version have been illustrated.

Many areas in Networker are still in need of further development. One of the most fundamental areas is ensuring that Networker includes typical browser features such as a Back, Forward and History button and a Status Bar. The evaluation of Networker sheds light on the necessity of an intelligent hyperlink connecting the WorkPad and the websites. The selection of this hyperlink from the WorkPad will allow the user to directly access the relevant website and vice versa. The author noted the importance of having a control in the WorkPad which can reference a website URL. For instance, when content from an airline website is copied across to the WorkPad, the 'Source' control is selected and then the URL of that website is displayed in the WorkPad area. Other areas for development in Networker in the future include, the management of pop-up windows, provision for overlapping windows that can be resized, and the provision for copying graphics and tables from the websites into WorkPad. Due to time constraints and the scope of the prototype these features were not developed.

Chapter 7: Conclusion

In this thesis the focus has been on information search on the web. This chapter summarizes the results of the thesis. It begins with a description of the goals of the study that have been met and the activities that the author conducted to meet the goals. The second section sheds light on the areas where Networker can be applied in industry and addresses the areas of Networker which can be developed in the future.

7.1 Discussion

The terms *browsing* and *surfing* imply a certain degree of aimlessness about the way users navigate the Web. Although we surmise that nearly all users, at least occasionally, allow serendipity to govern their Web exploration, many engage in a more purposeful behavior pattern called Collect-Compare-Filter-Process. NetWorker has been designed to provide a greater degree of support for the cycle than conventional browsers.

A number of goals were outlined in the beginning of this thesis.

- To identify user goals on the web and the tasks users complete to achieve these goals.
- To derive a process for decision-making on the web.
- To analyze current tools and determine to what extent they support user tasks.
- To develop key functions for a web support tool that overcomes the shortcomings of the current tools and assist users to deal with multiple web sites.
- To design and implement a web support tool, and address future development areas.

These goals will be discussed further in the sections below.

To Identify User Goals on the Web and the Tasks Users Complete to Achieve these Goals

Morrison et al.'s (2001) research determined that users *collect*, *find*, *explore* and *monitor* information on the web. This finding was the basis of this thesis and was used to create a decision-making framework.

To Derive a Process for Decision-making on the Web

A four-step decision-making cycle was defined beginning with users *collect*ing information, i.e. browsing from one web site to another and skim reading each site to find useful content, following this the users *compare* between useful and non-useful content of the collected web sites, next, users *filter* useful content into one workspace and finally, users *process* the content to meet their goal.

To Analyze Current Tools and Determine to What Extent they Support User Tasks

Commercial web tools such as, Web browsers, browser plug-ins, and window management systems and other research based web tools have been explored in great detail. Two groups of web support tools, which are most relevant to the decision-making framework, were reviewed. The first group consisted of two browsers Internet Explorer and Netcaptor, plus the Microsoft Window Management System. This group of tools supported the *Collect* and *Compare* steps of the framework. The second group consisted of two editing tools, Notes Pilot and Edit Pad. This group of tools supported the *Filter* and *Process* steps of the framework. The steps in the decision-making framework were broken down into low-level tasks and each tool was assessed against these tasks. The author made a qualitative assessment and the tools were placed into three categories: Poor Support, Good Support and Excellent Support. The compatibility between the tools was also assessed. This assessment was the pillar to finding the shortcomings of current web support tools and thus developing key functions for an integrated tool.

To Develop Key Functions for a Web Support Tool that Overcomes the Shortcomings of the Current Tools and Assist Users to Deal with Multiple Web Sites

The proposed tool retains the features of the current tools in the excellent category, enhances the features in the good category and makes provision for the features in the poor category. The literature reviewed in Chapter 2 also provided a basis for the functionality specified. New developments implemented include:

- allowing users to view content from more than one web site at any time;
- providing the user with a working area (a 'WorkPad') onto which web content can be pasted, URLs can be referenced, notes can be made, and calculations carried out;
- providing for user control of the number of windows to be displayed at any time, permit the user to switch between vertical and horizontal arrangement of windows, to suit the information being browsed
- providing for system placement of windows, with window size(s) being a function of the number of windows being displayed at any time, and the chosen orientation (horizontal or vertical).

The non-functional deliverables ensured that each task is conducted in a single operation; software is consistent to other software in the market and the system status is clear to the user and help is offered at all times.

To Design and Implement a Web Support Tool, and Address Future Development Areas

A number of design issues were addressed in Section 4.3 in Chapter 4 developing the proposed tool, Networker. The issues and the rationale have been discussed. In most cases, the best interests of the user have been considered. For instance, the user has the option of allowing one or two browser windows to be displayed in the tool leading to increased user flexibility and provide an effective way to compare web content from multiple web sites. Users also have the option to close the workspace created, WordPad or leave it open on its own, according to their preference. Similarly, the user

can choose which layout of browser window they prefer and interchange between horizontal and vertical and can choose between tabs and URL boxes to access new and previously visited websites. A number of implementation issues arose. Some of these were due to the limitations of Delphi, the software used to program Networker, and the web browser, Internet Explorer, used to access the websites and display in Networker. The support for opening multiple websites in each web browser and closing each individual web browser were two dilemmas that were resolved but took more time than expected. The functional requirements and non-functional requirements of Networker were all implemented other than the support for real-time user help and support.

A small-scale evaluation of NetWorker has produced findings:

1. First, users felt that the tool would benefit from a number of improvements - most of which have now been incorporated - to its interface and capabilities.

2. Secondly, the subjects felt quite strongly that NetWorker was unlike other tools they had used, and that they would use such a tool in their academic and personal activities. A tool such as NetWorker does not perform a hitherto impossible function. Rather, it exposes an approach to Web-based information processing that is supported by, but difficult to achieve with current tools.

7.2 Future Work

Networker is in a very early stage of development; it was intended to act as a prototype for experimenting with the look and feel of an interface paradigm combining multiple browser windows and a WorkPad. However, time for implementing the software was constrained and the interesting new functionality was implemented at the cost of a number of familiar browser and editor functions. Users tend to have a holistic approach to evaluating software and were, understandably, critical of these omissions. Accordingly, future versions will incorporate such obvious features as Back and Forward buttons for the browser, and methods to implement the

menu items shown in the WorkPad. A related problem concerned returning to pages that had previously been displayed in a browser tab; when a user opens a page in a NetWorker browser tab, then navigates to other pages by clicking on hyperlinks, the URL stored by the software for that page is not updated. Consequently, if the browser reloads, the page that appears is the one that was first loaded. Users found this particularly annoying, and in future versions; a history of URLs for the tab will be implemented. Thus the tab will be able to be reloaded with the page associated with the most recent URL, and back and forward.

Some problems with the interface remain; for example, some websites, when queried, produced results in popup windows. It will not necessarily be obvious to a naïve user that information can be dragged from a popup window into the WorkPad just as effectively as from a browser within NetWorker. Further, when information is dragged from a web browser into a RichEdit box, it sometimes displays as text, sometimes as a mixture of text and table, and sometimes as a mixture of body text, table, and HTML. A more reliable way of displaying the body text alone will need to be found. Moreover, the text will need to be supplemented with a web address in the WorkPad so that the user is aware of the reference. The user should be able to access the website from the web address in the WorkPad and access the WorkPad from the website.

Networker has a lot of potential as a commercial product. The evaluators, who tried the system using the web while doing assignments, unanimously agreed that it would be useful for academic or personal activities. It could be a fruitful tool for company-wide use in making strategic, marketing and economical decisions. Any decision a company makes needs to be justified after comparing multiple pieces of information. Examples of its use by average users have already been provided throughout the thesis.

References

Ackermann E.C. (1996) *Learning to the use the WWW*, Franklin Beedle and Associates, Inc, USA.

Air New Zealand (2003) *Air New Zealand Home*, Accessed March 2003 http://www.airnewzealand.com

Alexa (2003) *Alexa An Amazon.com Company,* Accessed March 2003 http://www.alexa.com

Apperley M.D., Mugridge R. and Hosking J. (1998) Tool integration, collaboration and user interaction issues in component-based software architectures. In Mingins, Christine and Bertrand Meyer (eds.) *Technology of Object-Oriented Languages and Systems: Proc TOOLS Pacific '98 Conference*, Melbourne, Australia, 1998, p 289—301.

Bass L. and Coutaz J. (1991) *Developing Software for the User Interface*, Addison Wesley, USA.

BayCon Group (1999) *Microsoft Windows 98 Manual,* Accessed October 2002 http://www.baycongroup.com/win98/microsoft_windows_98.htm#whattiles

Bhutan (2002) *University of Vienna, Department of Computer Science and Informatics Home page*, Accessed May 2002 http://www.cs.univie.ac.at/

Borland Software Corporation, Delphi Version 7.0 (Build4.453) (1983) *Borland Home*, Accessed October 2002, http://www.borland.com

Brandt D.S. (2000) *Evaluating Information on the Internet,* Accessed May 2000, http://thorplus.lib.purdue.edu/~techman/evaluate.htm

Brinck T., Gergle D. and Wood S. (2001) *Usability for the Web: Designing Web Sites that Work,* Morgan Kaufman, USA.

Byrne M., John B., Wehrle N. and Crow D. (1999) The tangled web we wove: A taskonomy of WWW use. *Proceedings of CHI'99 Conference on Human Factors in Computing Systems* Pittsburgh, May 15(20), pp. 544-551.

Brown M. and Shillner R. (1995) DeckScape: An Experimental Web Browser, Computer Systems and ISDN Systems*: Proceedings of the Third International World Wide Web Conference*. 10--14 April, Darmstadt, Germany, Vol. 27.

Card S. K., Robertson G.G. and York W. (1996) The WebBook and the Web Forager: an information workspace for the World-Wide Web. *Proceedings of the Human Factors in Computing Systems Conference, CHI, 96*. 111-117.

CIAWorld Fact book (2003) *CIAWorld Fact book Home,* Accessed March 2004 http://www.cia.gov/cia/publications/factbook/

ClickZ Stats (2002) *ClickZ Stats Home*, Accessed March 2004 http://www.clickz.com/stats/

Cove J.F and Walsh B.C. (1988) Online text retrieval via browsing. *Information Processing and Management*, 24(1):31--37, 1988.

Delphi Borland USA (2002) *Borland Home*, Accessed October 2002 www.borland.com/products/delphi/

Dillon A. and Vaughan M.W. (1997) It's the Journey & the Destination': shape and the emergent property of genre in evaluating digital documents. *The New Review of Hypermedia and Multimedia 3*: 91-106 (1997).

Dix A., Finlay J.E., Abowd G.E. and Beale R. (1998) *Human-Computer Interaction*. Second edition. Hemel Hempstead, Hertfordshire: Prentice Hall.

EditPad (2003) *Powerful, Versatile and Convenient Windows Text Editor to Edit All Kinds of Text Files*, Accessed September 2003. http://www.editpadpro.com/

Emarketer (2004) *Emarketer Home,* Accessed March 2004. http://www.emarketer.com

Freedom Air (2002) *Freedom Air Home*, Accessed September 2002. http:// www.freedomair.co.nz

Gerhard C. (2000) *The Internet Study,* Knowledge Networks group, Stanford Institute for Quantitative Study of Society, Accessed February 2002. http://www.stanford.edu/group/siqss/Press_Release/press_detail.html

Internet Explorer (2002) *Internet Explorer Home*, Accessed July 2002. http:// www.internetexplorer.com

Jacobson J. Q. (1999) *Delphi Developer's Guide to OpenGL*, Wordware Publishing Inc.

Johnson J. (2000), *Web Bloopers*, Morgan Kaufmann, USA

Johnson S. (2000), *American heritage Dictionary of the English Language Preface to Dictionary*, 4th edition, Accessed October 2003 http://www.bartleby.com/61/

Kaiser J. (2000) *Criteria for Web Site Evaluation: Points to consider when performing a site critique,* Accessed May 2002.
http://webdesign.about.com/library/weekly/aa071801a.html

King A.B. (2000) *What Makes a Great Web Site?* Accessed March 2002
http://webreference.com/greatsite.html

Kirakowski J. (1996) *Questionnaires in Usability Engineering. A List of Frequently Asked Questions* (3rd Ed.), Accessed August 2002.
http:// www.ucc.ie/hfrg/resources/qfaq1.html

Komando K. (2000a) *Beefing up your Web browser*, Accessed March 2002
http://www.usatoday.com/life/cyber/ccarch/2001/10/29/komando.htm

Komando K. (2000b) *Power shopping on the Internet,* Accessed March 2002
http://www.usatoday.com/life/cyber/ccarch/2001/10/29/komando.htm

Krug S. (2000) *Don't Make Me Think! A Common Sense Approach to Web Usability,* New Riders, USA, 2000.

Landsberger J. (2000) *Evaluating web site content.* Accessed March 2001
http://www.iss.stthomas.edu/webtruth/evaluate.htm

Lehnhert W.G. (1998) *Internet 101: A Beginner's guide to the Internet and the WWW,* Addison Wesley Longman Inc, Amherst, USA.

Lynch G. and Palmiter S. (2001) Design and Rapid Evaluation of Usable Web Sites, *CHI 2001 Tutorial*, ACM, USA, 2001.

Marchionini G. and Shneiderman, B. (1988) Finding facts vs. browsing knowledge in hypertext systems. *IEEE Computer* 19 (1), 70--80.

Microsoft Powerpoint (2002) *Microsoft Powerpoint Home,* Accessed July 2002.
http://www.microsoft.com/products/powerpoint/main.html

Microsoft Windows 98 (2002) *Microsoft Windows Home*, Accessed January 2002.
http://www.microsoft.com/windows/default.mspx

Microsoft Word (2002) *Microsoft Product Information Centre*, Accessed January 2002.
http://www.microsoft.com/products/

Morkes J. and Nielsen J. (1997) *Concise, Scannable and Objective: How to Write for the Web*. Accessed January 2002.
http://www.useit.com/papers/webwriting/writing.html

Morrison J.B., Pirolli P. and Card S.K. (2001) A Taxonomic Analysis of What World Wide Web Activities Significantly Impact People's Decisions and Actions, Interactive

poster, presented at the Association for Computing Machinery's *Conference on Human Factors in Computing Systems*, Seattle, March 31 - April 5, 2001.

Myers B.A. (1988) *A Taxonomy of User Interfaces for Window Managers*, IEEE Computer Graphics and Applications 8(5):65-- 84

Nahl D. and Paervaiz M. (1997) User-centred Assessment of Two Web Browsers: Errors, Perceived Self-Efficacy, and Success, *Proceedings of the 60th ASIS Annual Meeting*, Vol 34, Washington DC, Information Today, Inc (New Jersey), 1997

Netcaptor (2003) *Netcaptor Homepage*, Accessed September 2003.
www.netcaptor.com

Netscape 6.2 (2003) *Power Through More Choice*, Accessed September 2003.
http://home.netscape.com/browsers/6/choice.html

Nielsen J. (1993) *Usability Engineering,* Academic Press, New Riders, Indianapolis, USA.

Nielsen J. (1995) *Guidelines for Multimedia on the Web*, Accessed March 2002.
www.useit.com/alertbox/

Nielsen J. (1997a) *How people read on the Web*, Accessed March 2002.
www.useit.com/alertbox/

Nielsen J. (1997b) *Are Users Stupid?* Accessed March 2002.
www.useit.com/alertbox/

Nielsen J. (1997c) *The Tyranny of Bookmarks*, Accessed March 2002.
www.useit.com/alertbox/

Nielsen J. (1997d) *Jakob Neilsen on usability and web design*, Accessed March 2004.
http://www.useit.com/

Nielsen J. (1999) *Designing Web Usability: The Practice of Simplicity,* New Riders, USA.

Nielsen J. (2001) *The 3Cs of Critical Web Use: Collect, Compare, Choose*, Accessed April 2001.
http://www.useit.com/alertbox/20010415.html

Nielsen//NetRatings (2003) *The Global Standard for digital media measurement and analysis,* Accessed March 2004.
http://www.nielsen-netratings.com/news.jsp

Netcaptor, *Netcaptor Homepage*, Accessed April 2003
http://www.netcaptor.com/

Norman D.A. and Draper S.W. (1986) *User centered system design: New perspectives on human-computer interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Notes Pilot, *Two Pilots* (2003) Accessed Sept 2003.
www.surfpilot.net/npilot.html

Nunberg G. (Ed.). (1996) *The future of the book.* Berkeley, CA: University of California Press.

Pandia (2001) *On the Size of the World Wide Web*, Accessed March 2004.
http://www.pandia.com/sw-2001/57-websize.html

Phillips C.H.E., Mehandjiska D., Griffin D., Choi M.D. and Page D. (1998) The usability component of a framework for the evaluation of OO CASE tools, *Proc of Software Engineering: Education and Practice (SE:E&P'98),* IEEE, Dunedin, January 1998, 134-141.

Porteous M., Kirakowski J. and Corbett M. (1993) *SUMI User Handbook*, Human Factors Research Group, University College, Cork.

Preece J., Rogers Y., Sharp H., Benyon D., Holland S. and Carey T. (1994) *Human Computer Interaction Design*, Reading, Mass, Addison-Wesley.

Preece J., Rogers Y., and Sharp H. (2002*) Interaction design: beyond human computer interaction*, John and Wiley and sons Inc, New York

Rettig M. (1994) Prototyping for Tiny Fingers, *Comms of the ACM*, Vol 37, No 4, 21-28.

Schneiderman B. (1998a) *Designing the user interface: strategies for effective human computer interaction,* Addison Wesly Longman Inc., USA.

Schneiderman B. (1998b) *Elastic Windows for Rapid Multiple Window Management*, Accessed on Sept 2003.
www.cs.umd.edu/hcil/elastic-windows

Shneiderman B. and Kandogan E. (1997) Elastic Windows: evaluation of multi-window operations, *Proceedings of the SIGCHI conference on Human factors in computing systems*, p.250-257, March 22-27, United States

Spool J., and Czaja S. J. (2000) *The Future is Here, Tutorial Notes: Web Sites that Work, Designing with your Eyes Open*, User Interface Engineering, MA, USA.

Spool J., Scanlon T., Schroeder W., Snyder C., and DeAngelo T. (1999) *Web Site Usability: A Designers Guide*, Morgan Kaufmann, USA.

Tillman H. N. (2000) *Evaluating Quality on the Net,* Director of Libraries, Babson Park, Massachusetts, Accessed May 2003. http://www.hopetillman.com/findqual.html

Windows Developer Manual (1998) *Documentation*, Accessed August 2002. http://developer.apple.com/techpubs/mac/Toolbox/Toolbox-188.html

Windows 98 Introduction (1999) *Working with Multiple Windows*, Accessed August 2002.
http://www.uic.edu/depts/accc/seminars/win98-intro/mult-win.html

Appendix A: Program Code

```
unit Exp;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, OleCtrls, SHDocVw, ComCtrls, ToolWin,
  ActnCtrls, ImgList, ActnList, ActnMenus, math, extCtrls, Buttons,strUtils,
  ActnMan;

type
  TForm1 = class(TForm)
    addBrowserBtn: TButton;
    ActionManager1: TActionManager;
    Action1: TAction;
    Action2: TAction;
    Action3: TAction;
    Action4: TAction;
    Action5: TAction;
    Action6: TAction;
    Action7: TAction;
    Action8: TAction;
    Action9: TAction;
    Action10: TAction;
    Action11: TAction;
    Action12: TAction;
    Action13: TAction;
    ImageList2: TImageList;
    URLsMemo: TMemo;
    tileRightBitBtn: TBitBtn;
    tileDownBitBtn: TBitBtn;
    CloseBitBtn: TBitBtn;
    titlesMemo: TMemo;
    Timer1: TTimer;
    Timer2: TTimer;
    workPadPanel: TPanel;
    workPadRichEdit: TRichEdit;
    workpadLabel: TLabel;
    Action14: TAction;
    Action15: TAction;
    Action16: TAction;
    Action17: TAction;
    Action18: TAction;
    ActionMainMenuBar1: TActionMainMenuBar;
    workpadMinimaxBtn: TSpeedButton;

    procedure FormCreate(Sender: TObject);
    procedure CloseButtonClick(Sender: TObject);
    procedure addBrowserBtnClick(Sender: TObject);
    procedure DeleteBrowserBtnClick(Sender: TObject);
    procedure FormResize(Sender: TObject);
    procedure tileRightBitBtnClick(Sender: TObject);
    procedure tileDownBitBtnClick(Sender: TObject);
    procedure CloseBitBtnClick(Sender: TObject);
    procedure toggleUserInputBtnClick(Sender: TObject);
```

```
    procedure Timer1Timer(Sender: TObject);
    procedure workpadMinimaxBtnClick(Sender: TObject);
    procedure layoutWorkPad;

  private
    { Private declarations }
  public
    { Public declarations }
  end;
type tPositionParameterName = ( setTop,   setBottom, setLeft, setRight
                        , setWidth, setHeight
                        );

type tURLTitlePair = class(tObject)
  URL, title : string;
  constructor create;
  end; {tURLTitlePair}

type tURLsAndTitles = class(tList) // of tURLTitlePair
    function URLPos(URL : String) : integer;
    end;

type tjaBrowser = class(tTabControl)
    browserWindow   : tWebBrowser;
    URLBox          : tComboBox;
    closeButton     : tBitBtn;
    constructor create(owner :tComponent);
    procedure editKeyUp(Sender: TObject; var Key: Word; Shift: TShiftState);
    procedure display(newTop, newLeft, newHeight, newWidth : integer);
    procedure displayCloseButton;
    procedure displayURLBox ;
    function browserEncloses(X, Y : integer) : Boolean;
    procedure changeWebPage (Sender : tObject);
    procedure recordURL(URL : string);
    procedure downloadComplete(Sender: TObject; const pDisp: IDispatch; var URL: OleVariant);
    procedure updateTabs    (selectedTab : integer) ;
    procedure deleteBrowser(Sender: TObject);
private
    selectedTab     : integer;
    markedForDestruction : Boolean;
    end;

type tjaBrowserList = class(tList)
  displayDirection   : (LToR, down);
  constructor create;
  procedure display;
  function lastDisplayedBrowser : tjaBrowser;
  end; {class jaBrowserList}

const standardGap  = 5;
     panelColor   = $00cccccc;
     formColor    = $00aaaaaa;

var
  Form1: TForm1;
  var browsers      : tjaBrowserList; // of tWebBrowser pointers
  selectedBrowser   : integer;
  noOfBrowsers      : integer = 0;
  URLsAndTitles     : tURLsAndTitles;
```

```
  closeButtonBitmap : tBitmap;
  workPadPanelExpanded : Boolean = TRUE;


implementation

{$R *.dfm}
{=============================================================================
====}
procedure setPosition( var this : tControl;
                       pn1 : tPositionParameterName;
                       p1  : integer;
                       pn2 : tPositionParameterName;
                       p2  : integer;
                       pn3 : tPositionParameterName;
                       p3  : integer;
                       pn4 : tPositionParameterName;
                       p4  : integer
                     );
{=============================================================================
====}
var right, bottom                : integer;
var knowTop,  knowBottom, knowHeight,
   knowLeft, knowRight,  knowWidth   : Boolean;
begin
this.hide;
knowTop    := FALSE;
knowBottom := FALSE;
knowHeight := FALSE;
knowLeft   := FALSE;
knowRight  := FALSE;
knowWidth  := FALSE;

if pn1 = setTop    then begin this.top    := p1; knowTop    := TRUE; end;
if pn2 = setTop    then begin this.top    := p2; knowTop    := TRUE; end;
if pn3 = setTop    then begin this.top    := p3; knowTop    := TRUE; end;
if pn4 = setTop    then begin this.top    := p4; knowTop    := TRUE; end;

if pn1 = setHeight  then begin this.height  := p1; knowHeight := TRUE; end;
if pn2 = setHeight  then begin this.height  := p2; knowHeight := TRUE; end;
if pn3 = setHeight  then begin this.height  := p3; knowHeight := TRUE; end;
if pn4 = setHeight  then begin this.height  := p4; knowHeight := TRUE; end;

if pn1 = setBottom  then if knowTop    then this.height  := p1 - this.top
             else if knowHeight then this.top     := p1 - this.height;
if pn2 = setBottom  then if knowTop    then this.height  := p2 - this.top
             else if knowHeight then this.top     := p2 - this.height;
if pn3 = setBottom  then if knowTop    then this.height  := p3 - this.top
             else if knowHeight then this.top     := p3 - this.height;
if pn4 = setBottom  then if knowTop    then this.height  := p4 - this.top
             else if knowHeight then this.top     := p4 - this.height;


if pn1 = setLeft    then begin this.left    := p1; knowLeft   := TRUE; end;
if pn2 = setLeft    then begin this.left    := p2; knowLeft   := TRUE; end;
if pn3 = setLeft    then begin this.left    := p3; knowLeft   := TRUE; end;
if pn4 = setLeft    then begin this.left    := p4; knowLeft   := TRUE; end;

if pn1 = setWidth   then begin this.width   := p1; knowWidth  := TRUE; end;
```

```
if pn2 = setWidth   then begin this.width   := p2; knowWidth  := TRUE; end;
if pn3 = setWidth   then begin this.width   := p3; knowWidth  := TRUE; end;
if pn4 = setWidth   then begin this.width   := p4; knowWidth  := TRUE; end;

if pn1 = setRight   then if knowLeft  then this.width   := p1 - this.left
              else if knowWidth then this.left    := p1 - this.width;
if pn2 = setRight   then if knowLeft  then this.width   := p2 - this.left
              else if knowWidth then this.left    := p2 - this.width;
if pn3 = setRight   then if knowLeft  then this.width   := p3 - this.left
              else if knowWidth then this.left    := p3 - this.width;
if pn4 = setRight   then if knowLeft  then this.width   := p4 - this.left
              else if knowWidth then this.left    := p4 - this.width;

this.Show;
end;

{=============================================================================
====}
      procedure tjaBrowser.updateTabs(selectedTab : integer);
{=============================================================================
======}
var i : integer;
var browserNo : integer;
begin
// add the title to all the browsers' tabs; if no title, use the URL
if browsers <> nil then
 for browserNo := 0 to browsers.count-1 do
  begin
  tjaBrowser(browsers[browserNo]).tabs.Text := '';
  if URLsAndTitles <> nil then
   begin
   for i := 0 to URLsAndTitles.count-1 do
    if tURLTitlePair(URLsAndTitles[i]).title <> '' then
     tjaBrowser(browsers[browserNo]).tabs.add(tURLTitlePair(URLsAndTitles[i]).title)
    else
     tjaBrowser(browsers[browserNo]).tabs.add(tURLTitlePair(URLsAndTitles[i]).URL);
   //select the right tab....
   tabIndex:= selectedTab;
   end;
  end;
end; {of tjaBrowser.updateTabs}

{=============================================================================
======}
            procedure tjaBrowser.downloadComplete
     (Sender: TObject;const pDisp: IDispatch; var URL: OleVariant);
{=============================================================================
======}
type tSearchStatus = (searching, succeeded, failed);
var searchStatus : tSearchStatus;
var i          : integer;
var targetEntry  : integer;
var browserNo    : integer;
begin
// find the URL in the list
targetEntry :=  URLsAndTitles.URLPos(URLBox.text );

//add the page title to the title field of the target entry
if targetEntry = -1 then
```

```
  showmessage('URL "' + URLBox.text{browserWindow.locationURL} + '" is missing from URL list')
else
 tURLTitlePair(URLsAndTitles[targetEntry]).title := browserWindow.locationName ;

updateTabs(selectedTab);

end;

{=============================================================================
======}
       procedure tjaBrowser.changeWebPage (Sender : tObject);
{=============================================================================
======}
begin  // invoked when the user clicks on a browser's tab
browserWindow.navigate(tURLTitlePAIR(URLsAndTitles[tabIndex]).URL);
URLBox.text := tURLTitlePAIR(URLsAndTitles[tabIndex]).URL;
selectedTab := tabIndex;
end; {of procedure changePage (Sender : tObject)}

{=============================================================================
======}
      function tjaBrowserList.lastDisplayedBrowser : tjaBrowser;
{=============================================================================
======}
var i : integer;
candidateBrowser : tjaBrowser;
begin
if self <> nil then
 begin
 for i := 0 to count-1 do
  candidateBrowser := tjaBrowser(browsers[i]);

 lastDisplayedBrowser := candidateBrowser;
 end
else
 lastDisplayedBrowser := nil;
end; {of function tjaBrowserList.lastDisplayedBrowser : tjaBrowser}

{=============================================================================
======}
             constructor tURLTitlePair.create;
{=============================================================================
======}
begin
inherited create;
end; {of constructor tURLTitlePair.create}

{=============================================================================
======}
      function tURLsAndTitles.URLPos(URL:string) : integer;
{=============================================================================
======}
var i : integer;
position : integer;
var rightPartOfString : string;
begin
position := -1;
if URLsAndTitles <> nil then // there's something to search in
  for i := 0 to URLsAndTitles.count-1 do
```

```
    begin
    if ansiEndsText({subtext} tURLTitlePair(URLsAndTitles[i]).URL
              , {text}  URL) then
     position := i;
    end;

URLPos := position;
end; {of procedure URLPos(URL:string)}

{=============================================================================
======}
            procedure tjaBrowser.recordURL(URL : string);
{=============================================================================
======}
var i : integer;
var URLIsNew : Boolean;
var newURLTitlePair : tURLTitlePair;
begin
URLIsNew := TRUE;// until proven otherwise

// search the list of URLs to see if it's already there
if URLsAndTitles.URLPos(URLBox.text) = -1 then // the URL is not in the list
 begin
 newURLTitlePair := tURLTitlePair.create;
 newURLTitlePair.URL := URL;
 // We can't add the title yet, because it won't be available till the page has loaded
 // We'll use a documentLoaded handler to do that, when the time comes
 newURLTitlePair.title := ''; // in the meantime
 URLsAndTitles.Add(newURLTitlePair);
 end;

end; {of procedure tjaBrowser.recordURL(URL : string)}

{=============================================================================
======}
              procedure tjaBrowser.editKeyUp
      (Sender: TObject; var Key: Word; Shift: TShiftState);
{=============================================================================
======}
var i : integer;

var message : string;
begin
if key = vk_return then
 begin
 recordURL(URLBox.text);
 browsers.display;
 browserWindow.navigate(URLBox.text);

 selectedTab :=  URLsAndTitles.URLPos(URLBox.text);

 // if URLsMemo lacks the new URL, then add it to all the browsers' URLBoxes
 if form1.URLsMemo.lines.indexOf(URLBox.text) = -1 then
  begin
  form1.URLSmemo.lines.add( URLBox.text);
  for i := 0 to browsers.count-1 do
    if
tjaBrowser(browsers[i]).URLBox.items.indexOf(form1.URLsMemo.lines[form1.URLsMemo.lines.cou
nt-1])
```

```
      = -1 then
     begin

tjaBrowser(browsers[i]).URLBox.items.add(form1.URLsMemo.lines[form1.URLsMemo.lines.count-
1]);
     end;
  end;
 end;

end; {of procedure tjaBrowser.editKeyUp}

{=============================================================================
======}
    function tjaBrowser.browserEncloses(X, Y : integer) : Boolean;
{=============================================================================
======}
var encloses : boolean;
begin
Encloses :=  (X >= 0) and (X <= width)
     and (Y >= 0) and (Y <= height);
browserEncloses := encloses;
end; { of function tjaBrowser.browserEncloses}

{=============================================================================
======}
        constructor tjaBrowser.create(owner : tComponent);
{=============================================================================
======}
var i : integer;
begin
//create the tabWindow
inherited create(owner);
visible     := FALSE;
//multiLine    := TRUE;
raggedRight  := FALSE;
tabWidth     := 80;
parent       := form1;
color        := panelColor;
onChange     := changeWebPage;

//create the URLBox
URLBox       := tComboBox.create(owner);
URLBox.parent  := self;
URLBox.onKeyUp := editKeyUp;

//load the URLsBox from the master list of URLS in the URLsMemo
if form1.URLsMemo.lines.count > 0 then
 for i := 0 to form1.URLsMemo.lines.count - 1 do
  URLBox.items.append(form1.URLsMemo.lines[i]);

//create the browser window
browserWindow                := tWebBrowser.create(owner);
tControl(browserWindow).parent  := self;
browserWindow.OnDocumentComplete := downloadComplete;

//create the close button
closeButton                 := tBitBtn.create(owner);
closeButton.parent          := self;
closeButton.glyph.loadFromFile('smallCloseButton.bmp');
```

```
closeButton.layout          := blGlyphRight;
//closeButton.Caption         := 'Close this browser';
closeButton.OnClick         := deleteBrowser;

markedForDestruction := FALSE;

// update the tabs of the new browser from the URLsAndTitles list
end; {constructor tjaBrowser.create}

{=============================================================================
======}
            constructor tjaBrowserList.create;
{=============================================================================
======}
begin
inherited create;
end; {constructor tjaBrowserList.create}

{=============================================================================
======}
      procedure tjaBrowser.DeleteBrowser(Sender: TObject);
{=============================================================================
======}
var jaBrowser : tjaBrowser;
var i : integer;
begin
markedForDestruction := TRUE;

{ the statement above, (and a timer procedure) replace the code below, which
 doesn't work. Although the call to self.free executes OK, this method
 generates an "abstract error" when it returns, presumably because the object
 that the method belongs to no longer exists. So, instead,  a global timer that
 fires every 1/10s is used to clear away any browsers that have
 "markedForDestruction" set to TRUE. It's a hack, but it seems to work reliably.

if browsers.count > 0 then
 begin
 for i := 0 to browsers.count-1 do
  if browsers[i] = self then browsers.delete(i);
 self.free;
 browsers.display;
 end;
form1.Resize;
if browsers.count < 2 then form1.addBrowserBtn.Enabled := TRUE;
if browsers.count < 1 then form1.deleteBrowserBtn.enabled := FALSE;
}
end; {of procedure tjaBrowser.DeleteBrowser(Sender: TObject)}

{=============================================================================
======}
          procedure tjaBrowser.displayURLBox;
{=============================================================================
======}
begin
setPosition( tControl(URLBox)
     , setTop    , standardGap
     , setLeft   , standardGap
     , setWidth  , URLBox.parent.Width - CloseButton.width - 3 * standardGap
     , setHeight , 7 * standardGap);
```

```
end; {of procedure tjaBrowser.displayURLBox(jaBrowser : tjaBrowser; top : integer)}

{=============================================================================
======}
            procedure tjaBrowser.displayCloseButton;
//        ( newTop, newLeft, newHeight, newWidth : integer);
{=============================================================================
======}
begin
setPosition( tControl(closeButton)
      , setLeft   , clientRect.Right - 120 - standardGap
      , setTop    , standardGap
      , setWidth  , 120
      , setHeight , URLBox.Height
      );
end; {of displayCloseButton}

{=============================================================================
======}
                procedure tjaBrowser.display
       ( newTop, newLeft, newHeight, newWidth : integer);
{=============================================================================
======}
begin
setPosition( tControl(self)
        , setTop    , newTop
        , setLeft   , newLeft
        , setWidth  , newWidth
        , setHeight , newHeight
        );
tabHeight := standardGap * 3;

displayCloseButton;

displayURLBox;

updateTabs(selectedTab);

if URLsAndTitles <> nil then
if URLsAndTitles.count = 0 then // there should be no tabs
 setPosition( tControl( URLBox)
         , setLeft,    standardGap
         , setTop,     standardGap
         , setHeight,  URLBox.height
         , setRight,   closeButton.left - standardGap
         )
else // need to allow space for the tabs
 setPosition( tControl( URLBox)
         , setLeft,    standardGap
         , setTop,     standardGap
                 + abs(self.Font.Size) * self.Font.PixelsPerInch div 72
                 + 2 * standardGap
         , setHeight,  URLBox.height
         , setRight,   closeButton.left - standardGap
         );

setPosition( tControl(browserWindow)
        , setTop    , URLBox.BoundsRect.bottom + standardGap
        , setheight , height - URLBox.boundsRect.bottom - 2 * standardGap
```

```
        , setLeft   , standardGap
        , setWidth  , width - 2 * standardGap
        );

setPosition( tControl( closeButton)
        , setTop    , URLBox.top
        , setHeight , closeButton.height
        , setLeft   , closeButton.left
        , setWidth  , closeButton.Width
        );
end; {of procedure tjaBrowser.display(size : maxOrMin; top, left : integer)}

{=============================================================================
======}
                procedure tjaBrowserList.display;
{=============================================================================
======}
var i            : integer;
   j           : integer;
heightForBrowsers : integer;
widthForBrowsers  : integer;
heightperBrowser  : integer;
widthPerBrowser   : integer;
browserListLeft   : integer;
browserListTop    : integer;
browserTop        : integer;
browserHeight     : integer;
browsersToShow    : integer;
browserToShow     : integer;
begin
if browsers <> nil then
if browsers.count > 0 then
 begin

 if browsers.displayDirection = LToR then
  begin
  if workpadPanelExpanded then
   heightForBrowsers := form1.clientHeight div 2
                - (form1.tileDownBitBtn.Top + form1.tileDownBitBtn.height)
                - 2 * standardGap
  else // displayDirection = LToR and the workpadPanel is not expanded
   heightForBrowsers := form1.clientHeight
                - (form1.tileDownBitBtn.Top + form1.tileDownBitBtn.height)
                - 2 * standardGap;
  widthForBrowsers  := form1.width - 2 * standardGap;
  if browsers.count > 0 then
   widthPerBrowser   := widthForBrowsers div browsers.count - standardGap;
  heightPerBrowser  := heightForBrowsers;
  browserListLeft   := standardGap;
  browserListTop    := form1.tileRightBitBtn.top + form1.tileRightBitBtn.height + standardGap;
  end
 else //going down
  begin
  heightForBrowsers := form1.clientHeight
                - (form1.addBrowserBtn.Top + form1.addBrowserBtn.height)
                -  standardGap;
  browserListLeft   := standardGap;
  if workPadPanelExpanded then
   widthForBrowsers  := form1.clientWidth  div 2 - browserListLeft
```

```
                - standardGap
  else widthForBrowsers := form1.clientWidth - browserListLeft
                - standardGap;
  if browsers.count > 0 then
   heightPerBrowser:= heightForBrowsers div browsers.count
              - standardGap;
  browserListTop   := form1.tileRightBitBtn.top + form1.tileRightBitBtn.height + standardGap;
  widthPerBrowser  := widthForBrowsers;
  end;

 //Display the browsers.
 browsers.pack;
 if browsers <> nil then
 if browsers.count > 0 then
  for i := 0 to  browsers.count-1 do
   begin
   tjaBrowser(browsers[i]).tabs.clear;
   for j := 0 to browsers.count-1 do
    tjaBrowser(browsers[i]).tabs.add(tjaBrowser(browsers[i]).browserWindow.locationName);
   tjaBrowser(browsers[i]).display( {top}    browserListTop
                        , {left}   browserListLeft
                        , {height} heightPerBrowser
                        , {width}  widthPerBrowser);

   if browsers.displayDirection = LToR then
    browserListLeft := browserListLeft + widthPerBrowser + standardGap
   else // main browser display is downwards
    browserListTop := browserListTop + heightPerBrowser + standardGap;

   end;
 end;
end; { of procedure tjaBrowserList.display}

{=============================================================================
=======}
        procedure TForm1.FormCreate(Sender: TObject);
{=============================================================================
=======}
var I : integer;
var newBrowser : tWebBrowser;
begin
color := clMoneyGreen;
workPadPanel.color := panelColor;
workpadLabel.caption := 'Work Pad'; ////get rid of this if the appearance of the interface works

setPosition( tControl (tileDownBitBtn)

       , setLeft   ,   standardGap
       , setWidth  , 6 * standardGap
       , setTop    ,   standardGap
       , setHeight , 6 * standardGap
       );

setPosition( tControl (tileRightBitBtn)

       , setHeight , 6 * standardGap
       , setLeft   ,   tileDownBitBtn.Left + tileDownBitBtn.Width + standardGap
       , setWidth  , 6 * standardGap
       , setTop    ,   standardGap
```

```
        );

setPosition( tControl (AddBrowserBtn)

        , setLeft    ,    tileRightBitBtn.Left + tileRightBitBtn.Width + standardGap
        , setTop     ,    standardGap
        , setHeight  , 6 * standardGap
        , setWidth   ,    addBrowserBtn.width
        );

setPosition( tControl (closeBitBtn)
        , setWidth   , 15* standardGap
        , setLeft    ,    form1.clientRect.right - 16 *  standardGap
        , setTop     ,    standardGap
        , setHeight  , 6 * standardGap
        );
closeBitBtn.caption      :=    'Close';
closeBitBtn.show;

browsers := tjaBrowserList.create;
browsers.displayDirection := down;
selectedBrowser := 0;

URLsMemo.visible := FALSE; //the URLs Memo is the master list of URLs
URLsMemo.text := '';

addBrowserBtn.click;

//display this browser
tileDownBitBtn.click;
browsers.display;

URLsAndTitles := tURLsAndTitles.create;

layoutWorkPad;
form1.caption := 'NetWorker';

end; {of form.create}

{=============================================================================
=======}
        procedure TForm1.CloseButtonClick(Sender: TObject);
{=============================================================================
=======}
begin
close;
end;

{=============================================================================
=======}
       procedure TForm1.addBrowserBtnClick(Sender: TObject);
{=============================================================================
=======}
var newBrowser : tjaBrowser;
begin

// create a new browser
NewBrowser := tjaBrowser.Create(Form1);
tControl(NewBrowser). parent := form1;
```

```
//add the new browser to the master list of browsers
browsers.insert(0,newBrowser);

//put the existing list of URLs into the new browser's URLBox
if browsers.count > 1 then
 if tjaBrowser(browsers[1]).URLBox.items.text <> '' then
  newBrowser.URLBox.Items.text := tjaBrowser(browsers[1]).URLBox.items.text;

newBrowser.updateTabs(1);

//display all the browsers
browsers.display;

if browsers.count > 1 then addBrowserBtn.Enabled := FALSE;

layoutWorkPad;
end; {of procedure TForm1.addBrowserBtnClick(Sender: TObject)}

{=============================================================================
======}
      procedure TForm1.DeleteBrowserBtnClick(Sender: TObject);
{=============================================================================
======}
var jaBrowser : tjaBrowser;
begin
if browsers.count > 0 then
 begin
 jaBrowser := browsers[0];
 browsers.delete(0);
 jaBrowser.destroy;
 browsers.display;
 end;
layoutWorkPad;
if browsers.count < 2 then addBrowserBtn.Enabled := TRUE;
end; {of procedure TForm1.DeleteBrowserBtnClick(Sender: TObject)}

{=============================================================================
======}
            procedure tForm1.layoutWorkPad;
{=============================================================================
======}
begin
if workPadPanelExpanded then
 begin
 if browsers <> nil then
  if browsers.count = 0 then //there are no browsers; we maximise the workpad
   setPosition( tControl(workPadPanel)
          , setLeft   , standardGap
          , setTop    , addBrowserBtn.boundsRect.bottom + standardGap
          , setBottom , clientRect.Bottom - standardGap
          , setWidth  , clientRect.right - 2 * standardGap
          )
  else // there's at least one browser
   if browsers <> nil then
    if browsers.displayDirection = LToR then
     begin
     setPosition( tControl (workPadPanel)
            , setLeft    , standardGap
```

```
                , setRight   , clientRect.right - standardGap
                , setTop     , tjaBrowser(browsers[0]).boundsRect.bottom + standardGap
                , setBottom  , form1.clientRect.bottom  - standardGap
                );
        setPosition( tControl(workpadMinimaxBtn)
                , setRight, workpadPanel.boundsRect.right - standardGap
                , setWidth, workpadMinimaxBtn.width
                , setTop,   workpadMinimaxBtn.top
                , setheight,workpadMinimaxBtn.height
                );
        end
      else // displayDirection is down
        begin
        setPosition( tControl (workPadPanel)
                , setLeft   , tjaBrowser(browsers[0]).boundsRect.right + standardGap
                , setRight  , clientRect.right - standardGap
                , setTop    , addBrowserBtn.BoundsRect.bottom + standardGap
                , setBottom , clientRect.bottom - standardGap
                );
        end
  else // browsers = nil; we maximise the workpad
      setPosition( tControl(workPadPanel)
            , setLeft   , standardGap
            , setTop    , addBrowserBtn.Top + addBrowserBtn.Height + standardGap
            , setHeight , clientRect.Bottom - workPadPanel.Top - standardGap
            , setWidth  , clientRect.right - 2 * standardGap
            );
  end
  else // workPadPanelExpanded is FALSE
  begin
  setPosition( tControl (workpadPanel)
          , setLeft   , addBrowserBtn.BoundsRect.Right + standardGap
          , setRight  , closeBitBtn.left - 2 * standardGap
          , setHeight , addBrowserBtn.height
          , setTop    , standardGap
          );
  end;

setPosition( tControl(workpadMinimaxBtn)
          , setwidth  , 117
          , setHeight , 22
          , setRight  , workpadPanel.Width  - standardGap
          , setTop    , standardGap
          );

setPosition( tControl (workPadLabel)
          , setTop    , standardGap
          , setHeight , workpadLabel.height
          , setRight  , workpadMinimaxBtn.boundsRect.left - 10 * standardGap
          , setWidth  , workpadLabel.Width
          );

setPosition( tControl (workPadRichEdit)

          , setLeft   , standardGap
          , setWidth  , workPadPanel.Width - 2 * standardGap
          , setTop    , actionMainMenuBar1.BoundsRect.bottom + standardGap
          , setBottom , workPadPanel.boundsRect.bottom - 8* standardGap
          );
```

```
end; {of layoutWorkPad}

{=============================================================================
=======}
        procedure TForm1.FormResize(Sender: TObject);
{=============================================================================
=======}
begin
if browsers <> nil then
if browsers.count > 0 then
 browsers.display;
layoutWorkPad;
setPosition( tControl (closeBitBtn)
        , setRight, clientRect.right - standardGap
        , setWidth, closeBitBtn.width
        , setHeight, closeBitBtn.height
        , setTop, closeBitBtn.top
        );
end; {of procedure TForm1.FormResize(Sender: TObject)}

{=============================================================================
=======}
      procedure TForm1.tileRightBitBtnClick(Sender: TObject);
{=============================================================================
=======}
begin
browsers.displayDirection := LToR;
tileRightBitBtn.enabled := FALSE;
tileDownBitBtn.enabled  := TRUE;
browsers.display;
layoutWorkPad;
end;

{=============================================================================
======}
        procedure TForm1.tileDownBitBtnClick(Sender: TObject);
{=============================================================================
======}
begin
browsers.displayDirection := down;
tileRightBitBtn.enabled := TRUE;
tileDownBitBtn.enabled  := FALSE;
browsers.display;
layoutWorkPad;
end;

{=============================================================================
======}
        procedure TForm1.CloseBitBtnClick(Sender: TObject);
{=============================================================================
======}
begin
close;
end;

{=============================================================================
=======}
     procedure TForm1.toggleUserInputBtnClick(Sender: TObject);
```

```
{=========================================================================
=======}
begin
if workPadPanelExpanded then
  begin
  workPadPanelExpanded := FALSE;
  workPadPanel.Top := standardGap;
  workPadPanel.Height := workpadLabel.boundsRect.bottom + standardGap;
  workPadPanel.left   := clientRect.right div 2;
  workPadRichEdit.visible := FALSE;
  end
else // workPadPanelExpanded = FALSE
  begin
  workPadPanelExpanded := TRUE;
  workPadRichEdit.visible := TRUE;
  end;

layoutWorkPad;
browsers.display;
end; {of procedure TForm1.toggleUserInputBtnClick(Sender: TObject)}

{=========================================================================
=======}
procedure TForm1.Timer1Timer(Sender: TObject);
{=========================================================================
=======}
var i : integer;
type tSearchStatus = (searching, succeeded, failed);
var searchStatus : tSearchSTATUS;
begin
if browsers <> nil then
if browsers.count > 0 then
  begin
  searchStatus := searching;
  i := 0;
  while searchStatus = searching do
    begin
    if tjaBrowser(browsers[i]).markedForDestruction then
      begin
      tjaBrowser(browsers[i]).Free;
      browsers.delete(i);
      browsers.display;
      layoutWorkPad;
      searchStatus := succeeded;
      end;
    i := i + 1;
    if (searchStatus = searching)
    and (i > browsers.count-1) then
      searchStatus := failed;
    end;
  if browsers.count > 1 then addBrowserBtn.Enabled := FALSE;
  if browsers.count < 2 then addBrowserBtn.enabled := TRUE;
  end;

end;

{=========================================================================
=======}
      procedure TForm1.workpadMinimaxBtnClick(Sender: TObject);
```

```
{=========================================================================
======}
begin
if workPadPanelExpanded then
  begin //we prepare the panel to be collapsed
  workpadRichEdit.visible   := FALSE;
  workpadPanelExpanded      := FALSE;
 // workpadMinimaxBtn.Glyph.loadFromFile('restore.bmp');
  workpadMinimaxBtn.Caption := 'Show Workpad';
  end
else // workPadPanelExpanded = FALSE
  begin // we prepare the panel to be expanded
  application.processmessages;
  workPadRichEdit.visible := TRUE;
  workPadPanelExpanded := TRUE;
  //workpadMinimaxBtn.Glyph.loadFromFile('minimise.bmp');
  workpadMinimaxBtn.Caption := 'Hide Workpad';
  end;

browsers.display;
//here's where the expansion and collapsing will really take place
layoutWorkPad;

end;

end.
```

A.1 Driving Instructions

Step 1: Open the executable file (.exe) file in the Delphi folder

Step 2: The NetWorker tool is displayed, type in a URL in the combo box

Step 3: A dial-up connection dialogue box appears, type in user name and password then click on Connect button *(See Figure 7.1)*

Step 4: The web site will display in the browser window

Step 5: Type in other urls in the combo box to display new web sites

Step 6: To view previous sites click on the tabs that appear at the top of the browser window, each tab has a title of the respective website OR
click on the downward arrow in the combo box to view other url locations and click on any url to view the website in the browser window

Appendix B: Evaluation Documentation

B.1 Description of Networker

Before we begin the tasks, I'd like you to explore Networker independently for as long as five minutes. As you explore, please "think aloud". That is, please tell me your thoughts as you encounter the different features of Networker. Feel free to explore the content on the Web that is of interest to you. If you complete your independent exploration before the five minutes are up, please let me know and we'll proceed on with the tasks. Again, please remember to tell me what you're thinking as you explore Networker.

Figure B.1 is an illustration of the Networker tool. The browser window and workpad window has a vertically tiled layout. In the left side of the tool, there is a browser area. A maximum of two browser windows can be seen in this section. In the right side of the tool, there is a workpad section. A workpad is placed there. The purpose for this layout is so that users can collect web sites on the left side and drag across the content to the right side. Once the HTML content is in the WorkPad, the user will format the content and analyse it.
When the user selects the Horizontal layout button the browser window resizes to fit the whole top half of the tool and the workpad resizes to fit the bottom half of the tool. See Figure B.2.

A description of the components of Networker:

Browser Area
This is the region where browsers are displayed. The region is generally filled with one or two browsers. By default one browser is displayed.

Scroll Bar
A horizontal scroll bar is available on Networker. It can be scrolled from left to right.

Add browser button
This button adds new browser windows into the browser area. The browsers fit to the browser area and resizes according to the number of browsers the user prefers to use.

Horizontal Layout Button
Users can use this button to tile the browser window and workpad in a horizontal direction. This is used for websites that have a longer width so as more content can be seen. In this layout, web content will be copied down to the workpad.

Vertical Layout Button
Users can use this button to tile the browser window and workpad in a vertical direction. This is used for websites that have a longer height so as more content can be seen. In this layout, web content will be copied across to the workpad.

Close Browser button
This button closes individual browsers. Once the last browser window is closed, the workpad resizes to fit the display area (browser area and workpad area). If one browser window remains, the browser window will resize to fit the browser area.

Tabs
Tabs are used to reference all the web sites the user has visited and to access those web sites at any time.

Combo Box
Combo box is used in all browser windows. It allows the user to type in a URL and then saves it. The user can select a previously visited URL or type new ones in the combo box. The web sites are displayed in the browser window.

Workpad Area
This is where the workpad, and menu bar is placed. Web content can be copied into the workpad, arranged and formatted. Users can further analyse the accumulated content.

Workpad Menu Bar
This menu bar is still not implemented. There are four menus, File, View, Format and Help. File opens workpads, creates new workpads, saves workpads and prints workpad content. View allows users to view the tool in a vertical layout, horizontal layout and bi-focal layout. The bi-focal layout is still not implemented.

Minimise/RestoreWorkpad Button
This button is a toggle button that either minimises the entire workpad area or restores it to its actual size.

Close button
This button closes the Networker tool.

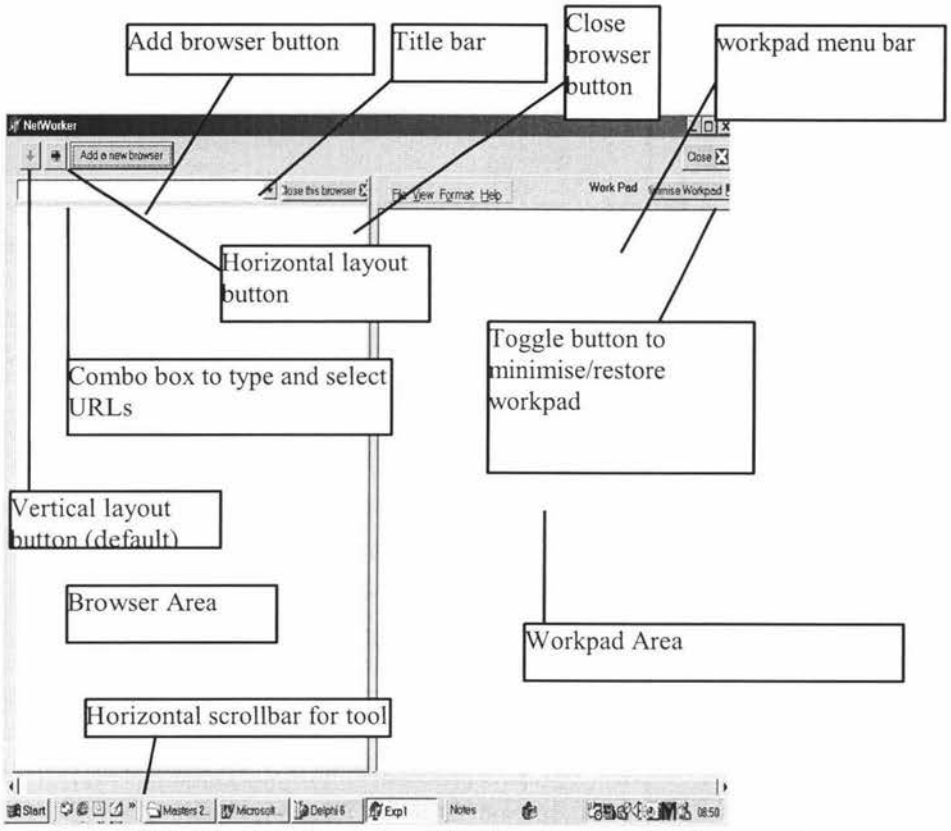

Figure B.1: Vertical layout of Networker tool using one browser window and workpad

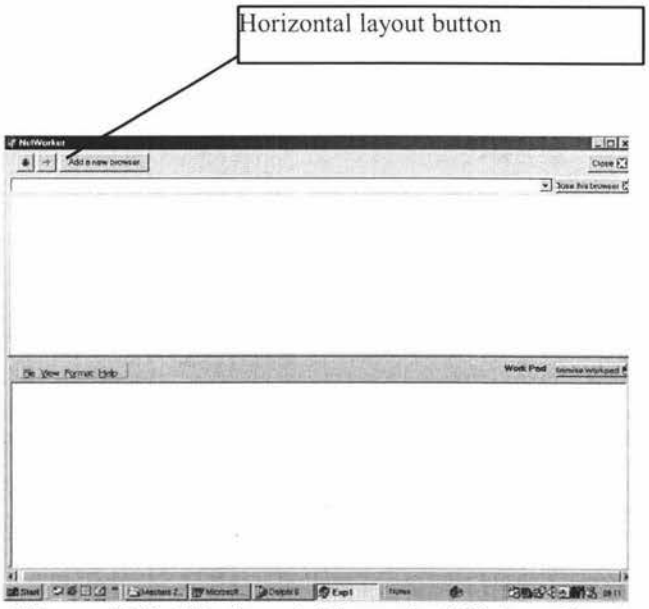

Figure B.2: Horizontal layout of Networker tool using one browser window and workpad

B.2 Information Sheet

Thank you very much for participating in this study. The goal of this project is to evaluate the interface of Networker. The results of the evaluation will be summarised and recorded in the Masterate thesis. You will be asked to use Networker to search for the best solution to common tasks Web users conduct on a regular basis. You will be asked to "think aloud" as you use the Networker tool. I'll need you to review and sign this statement of informed consent. Please let me know if you have any questions about it.

We'll start with an overview of Networker. It's a web cantered prototype developed for the purpose of completing the requirements of a Masterate thesis. Its purpose is to allow web users to collect and compare web content in an efficient and organised manner.

The purpose of our work today is to explore the Networker interface to identify features that could be improved. I'm also interested in finding out about features that are particularly helpful.

In a few minutes I'll give you three tasks. For each task you will use Networker to conduct information searches on the web and find an optimum solution to the tasks.

As you use Networker to find information for each task, please keep in mind that it is Networker that is the subject of the evaluation- not you.

You should feel free to work on each task at a pace that is normal and comfortable for you. I will be keeping track of how long it takes you to complete each task, but you should not feel rushed. If any task takes you longer than ten minutes, I will ask you to move on to the next task. I'll ask you to restart Networker before starting a new task.

There is no set solution to any of the tasks. But if you feel unable to complete a task and would like to stop, please say so and we'll move on to the next task.

Before we proceed, do you have any questions at this point?

Before we begin the tasks, I'd like you to explore Networker independently for as long as five minutes. As you explore, please "think aloud". That is, please tell me your thoughts as you encounter the different features of Networker. Feel free to explore the content on the Web that is of interest to you. If you complete your independent exploration before the five minutes are up, please let me know and we'll proceed on with the tasks. Again, please remember to tell me what you're thinking as you explore Networker.

B.3 Questionnaire

Please circle the most appropriate selection.

Age Range: 10-19 20-29 30-39 40-49 50-59 60+

Gender: Male Female

Occupation: Student Software industry Other

Internet/Web Experience:

Research, Information Gathering	Daily	Weekly	Monthly	Never
Bulletin Board Posting	Daily	Weekly	Monthly	Never
Chat Room Usage	Daily	Weekly	Monthly	Never

Please check the box to show your agreement or disagreement with the following statements:

Table B.1 Questionnaire

Question	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
It is clear what the purpose of the tool is					
It is clear how to use the tool					
It is easy to collect information using the tool					
It is easy to compare multiple web sites using the tool					
It is easy to refer back to visited web sites					
It is easy to copy important information and edit using the workpad					
It is easy to view specific information using the tool					
The work space of the tool can be customized to my needs					
The labels and icons on the buttons are clear to read					
The labels and icons on the buttons are clear to understand and use					
The tool will be useful to my academic or personal activities on the Net					
The design and layout presents an aesthetically pleasing manner					
I prefer the lighter background color than a darker colored background					
The tool is very similar to other tools I have used					

Please add any recommendations for changes to the overall design, language, controls of the tool on the back of this paper.

How did you feel about your performance on the tasks overall?
What was the best thing about the tool?
What was the worst thing about the tool?

Thanks for your participation in the testing of the screens-only prototype.

B.4 An illustration of Networker at work

Task
Jane is a tertiary student in Palmerston North. She wants to fly to Sydney in the Christmas holidays. She wants to fly on the 24th December 2002 and return on January 15th 2003. She is looking for the most competitive price in the market. How can Networker help Jane?

Step 1:
Jane types in the Air New Zealand web site URL in the combo box and presses 'enter' (See Figure B.3).

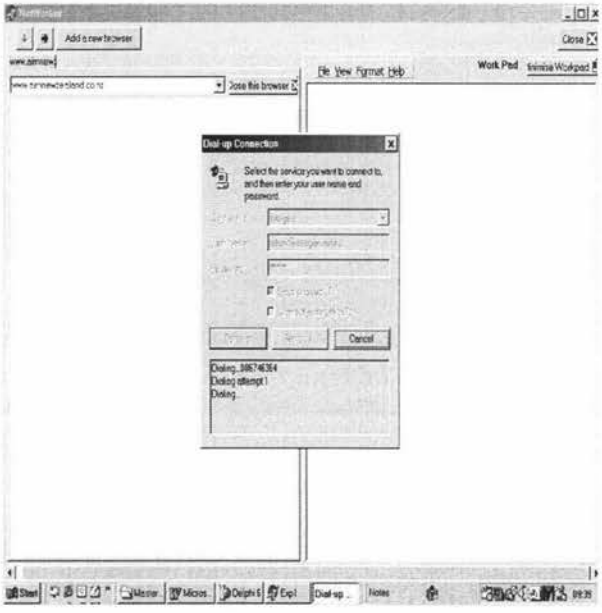

Figure B.3 Login and type in URL

Step 2:
Jane rearranges browser area so that the full web site can be seen. To do this she minimises the workpad area and presses the 'minimise workpad' button (See Figure B.4).

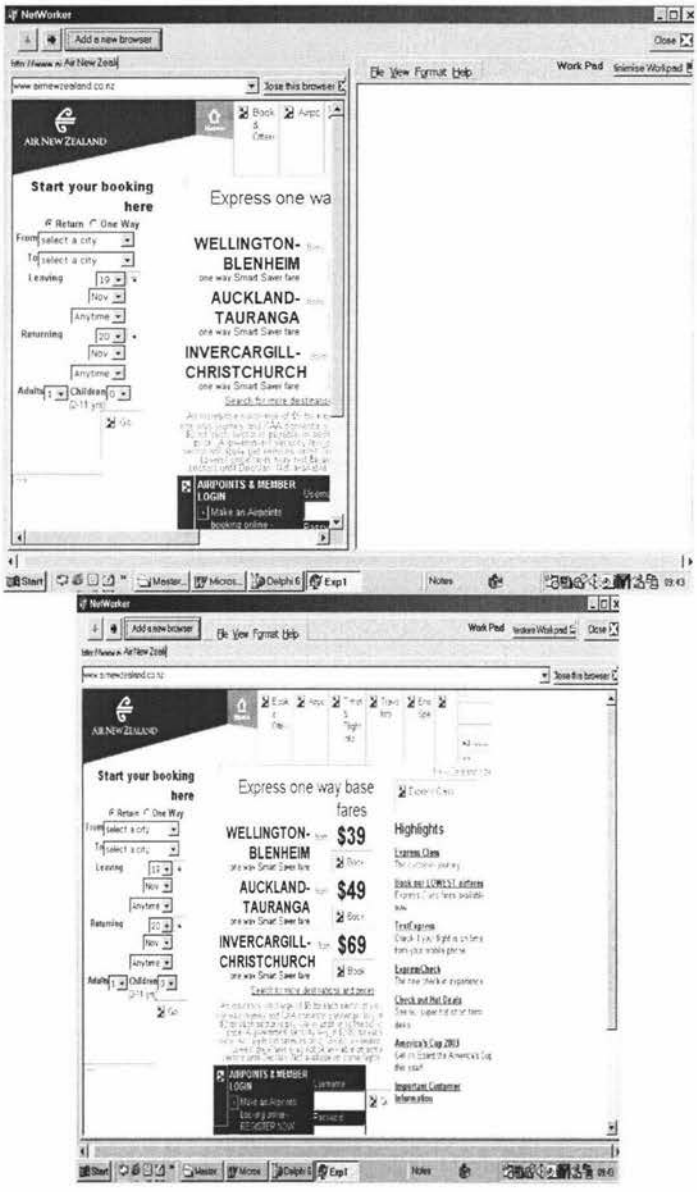

Figure B.4 Minimise workpad

Step 3:
Jane navigates to the international site by typing in the city names and dates for arrival and departure then restores the workpad area by clicking restore workpad area button and copies the itinerary and price quotes into the work area (See Figure B.5).

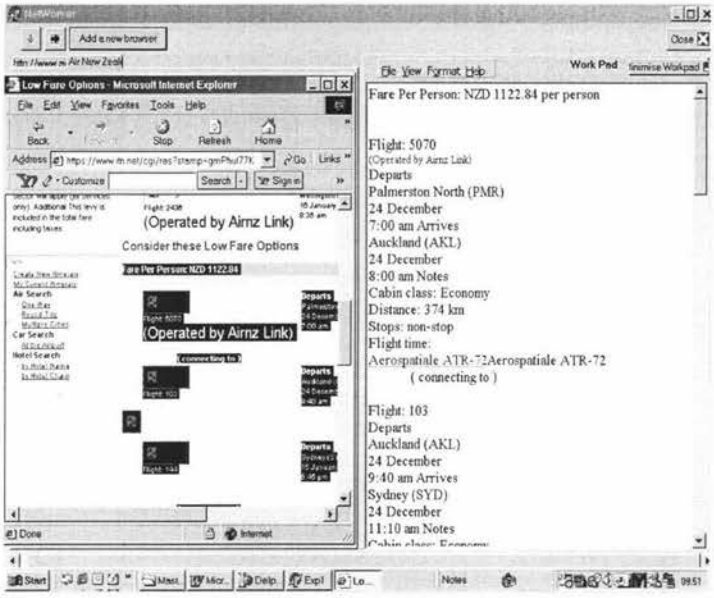

Figure B.5 Transfer content to Workpad

Step 4:
A new web site is opened (See Figure B.6)

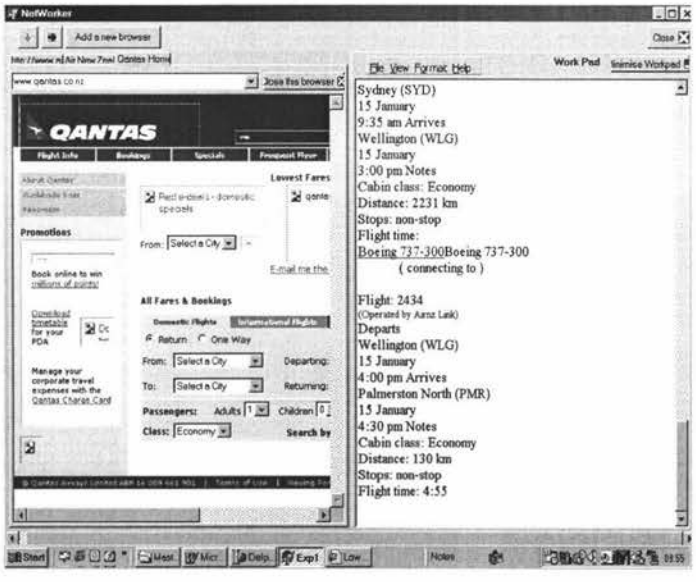

Figure B.6 Open web site

Step 5:
Jane navigates to find the itinerary and pricing information. Window is resized to see the whole web site (See Figure B.7)

Figure B.7 Resized window

Step 6:
Jane copies important itinerary and pricing content across to work pad (See Figure B.8).

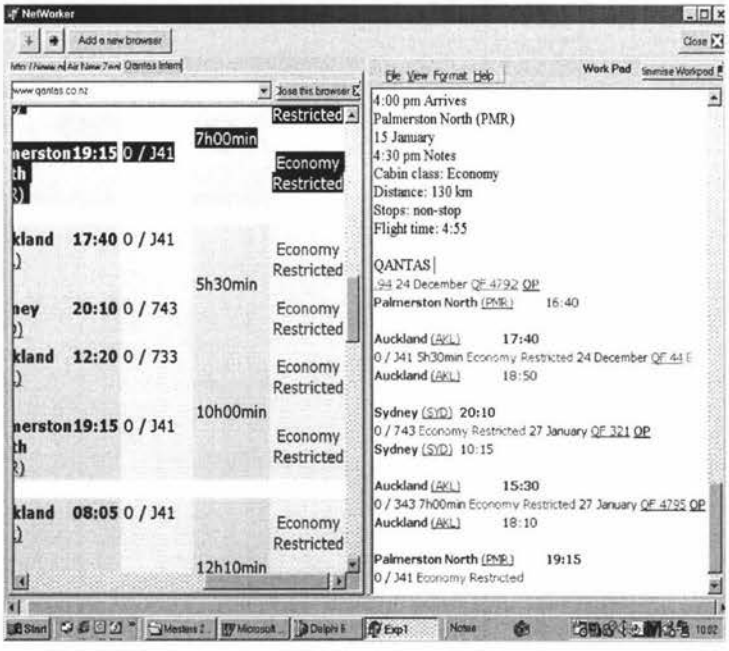

Figure B.8 Content transferred to Workpad

Step 7:
A third web site is opened.
Minimise workpad to view full screen
Copy important itinerary and pricing content across to workpad (See Figure B.9).

Figure B.9 Content transferred to Workpad

Step 8:
Resize workpad to analyse and format the information. Press the 'close this browser ' button (See Figure B.10).

Figure B.10 Format and Analyse content in Workpad

Step 9:

Jane decides to choose Freedom air as it has the most competitive rates during the time she wishes to travel. However, she needs to depart on December 25th and return on January 23rd 2002. Jane will now book a ticket. She needs to minimise the workpad and open the browser window and click on the Freedom Air tab to make a booking on the Freedom Air web site (See Figure B.11).

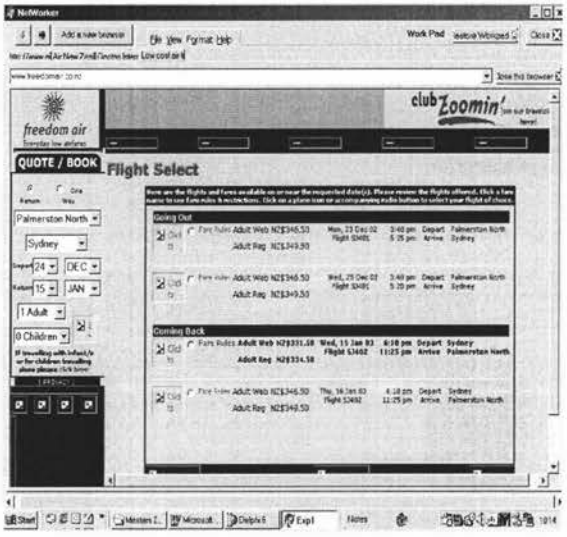

Figure B.11 Final selected web site

Appendix C : Evaluation Results

C.1 Transcriptions

Participant 1

Task A, B and C
I can't go back?
This is a very difficult interface
It is hard to grasp and apply
The tabs do not reload the old websites?
How come I can't close the tabs?
What do the arrows mean?
The arrows are not intuitive.
I prefer the vertical layout.
The hide and show workpad control is not intuitive.
It is not obvious that to resize a browser the workpad needs to be hidden
Dragging content across to workpad does not work.
Ok, it does work but this is difficult.
I thought that the cross by the browser window was for selecting and closing the tabs
sometimes tabs do not display related sites or any sites
spent too long collecting web sites could not reach compare stage
I'm so frustrated, time is running out and I have not completed most of the tasks

Participant 2

Task A
Handy to resize, drag workpad down
We need something that tells me that its working otherwise I have no idea what's going on
It (search engines) should recognise enter key , your browser might be doing that
should not have to log in every time, I always open in a new page so as to leave the searched one there
I hate buying things off the web, they don't even say what price its in
It should have taken me to a thing where I get to see the currency
Tried to copy paste, did not work I want to get that url into the workpad, not doing anything
I'll try the old fashioned way copy paste- there we go
It would be nice to minimise that workpad
But its annoying hiding and showing the workpad every time
There we go there's big error a delphi one - had to restart, url typing problem past the limit
Now I suppose the pad would have lost what was there
Oh it hasn't got a memory either

That's definitely something you'll need to see in the future
There we go it's asking me to log in again

Task B

I personally like the horizontal layout because I don't like scrolling across I'm old fashioned
Do you want combination of years or for me to pick one
I can't believe this one bank has 7 years the other has five years
Got another script were
Now normally want I'd want to do is go back to the BNZ site and look at their 5 year rate and keep a record
So how come it wen back to the old url in the combo box
I haven't said it yet
All I've done is typed in www
Just assumed has it
Where were we in the web site before
If I copy over on work pad will it overwrite that
No
In fact it lost the highlighting
It pasted it in them middle as I placed it
Memorized the first copy I did
What's happened to the BNZ tab/ first one?
Site location name page saved
Browser it memorizes password and as long as your using the proxy
Very close BNZ is .01 percent cheaper it will make a big difference I'm very sure
We finished yahoo, with a prototype its normal to not get finished

Task C

I just want rates, not all that info
There is actually a nz motel federation, don't know web sites, have an independent rating
Page not found
I opened a new window funny how it goes into IE why is it not opening up in your tool
Your window is a front end to IE
They are both asking you to log in
Wonder if you can copy more than one line
Losing my new windows that are not set in tool
Can do several lines
Copying in title and internet links, banners, images would be very handy
Copies links with page sources - very long and tedious
Colors shocking on site can hardly read that
IE pages and tools- should open new windows in your browser
Look at tabs

Participant 2 added a number of extra comments.

If the work pad is hidden it could be sized smaller.
Reloading each page from the cache makes the process slower.
When copying and editing behavior the tool did some unexpected behavior, did not always so as one expects in a "Windows" environment.
I could not resize the windows and overlaying windows to hide my work.
I like the clean and uncluttered appearance.
You can't judge the speed of the inputs as it depends on the network.
I am often grabbing pieces of paper to scribble down information off a web page before moving onto another page and the ability to simply highlight and drag the content to a work pad is very useful. Would like to be able to drag graphics as well as text to work pad.
I have never seen anything like this before.

Participant 3

Task A, B and C
Why didn't you stick to the normal Windows menu bar that we users are use to?
The menus need to be longer and if you show the interface with no explanations I might think I'm looking at a UNIX station. You should stick with a Microsoft interface running on a Microsoft platform.
About should mention version.
The interface functionality is very basic.
Why doesn't the browser have history, channels, bookmarks and navigation controls like IE and Netscape Navigator.
If I click on this link it should take me to some URL and show it as an extension in the URL box. If there are no extensions how should I bookmark the page?
If I cannot bookmark then to me the interface has limited functionality.
I want to select and drag the whole page.
I cannot scroll down on work pad when it is in horizontal layout.
I'll just do things the old fashioned way and use pen and paper.
Why do I need to log in each time?
Two browsers cram the work space. I will close one off.
Where are the back and forward buttons?
I will try showing workpad and dragging content, this time (Task B)
Used Ctrl C+ crtl V to switch between IE window and work pad (Task C)
Why are all new windows not opening in Networker browser?
I like yellow pages site I'll go to that one. Why is it calling on IE interface?

Participant 4

Task A
The cursor for workpad does not copy colored areas with white text as workpad has a white background.
Does not deselect dragged text.
One site does not have leather back will paper copy do?
Do you want me to get the price for that one?
How do I get back here?

I'll just type in the keywords.
I can try two browser windows.
I only need one open now.

Task B
Can I type inside the work pad?
The bank rates are all the same. That's very handy.
The most competitive rate is all of them.

Task C
I am looking for a motel in the heart of the city.
There is too less variety on this site, there must be a better way of finding a motel.
I want to copy the lettering (source) from the banner to the text. I can't do that, I'll just type it in.
The most competitive is Alglen Motel.

Participant 5

Task A
How do you get back?
So I can't just put "mazda" in the Url box
Kids books are loading
Go back to the search engine
Price is $45, show notepad, select and drag content across
Go to another web site
b+ n web site has the same price $45

Task B
Select a 5 year period
Add in work pad
Show workpad
Record bank rates, hide work pad and move on
Best rate is kiwibank 7.3

Task C
I'll go into yellow pages, it is a big web site and should have what I need
I'll go straight to accommodation, motels and lodges
I'll investigate Apollo motel, close to city so its fits the criteria
It has lots of features, is in the price range I'll record in work pad
Hide work pad and carry on
Close to city and airport
Show work pad and drag content
Alglen is the cheapest, $79

C.2 Questionnaire Analysis

Table C.1 Question 1

	Participant(s)				
Question	1	2	3	4	5
It is clear what the purpose of the tool is	agree	strongly agree	neutral	strongly agree	neutral
It is clear how to use the tool and move from one part of the task to another	disagree	agree	agree	strongly agree	agree
It is easy to collect and view web sites using the tool	disagree	agree	strongly disagree	strongly agree	agree
It is easy to compare multiple web sites using the tool	disagree	neutral	agree	strongly agree	agree
It is easy to refer back to visited web sites	strongly disagree	agree	strongly disagree	neutral	agree
It is easy to copy important information using the work pad	disagree	agree	strongly agree	agree	agree
It is easy to make the software do exactly what you want	disagree	neutral	neutral	strongly agree	neutral
The work space of the tool can be arranged to my needs	strongly disagree	disagree	agree	strongly agree	agree
The labels and icons on the buttons can be clearly read and are understandable	disagree	strongly agree	agree	strongly agree	agree
The tool has an attractive presentation	neutral	strongly agree	neutral	strongly agree	neutral
The software responds too slowly to inputs	neutral	neutral	disagree	strongly disagree	disagree
The tool will be useful to my academic and personal activities on the Net	strongly disagree	strongly agree	neutral	strongly agree	strongly agree
The tool is similar to other tools I have used	strongly disagree	strongly disagree	disagree	disagree	disagree

Question 2

What was the best thing about the tool? What were the most useful features of the tool?

Participant 1 believes that the option of using two browser windows was the best thing about the tool.

Participant 2 thought that the clean uncluttered interface is the best feature. The concept of the Work Pad is very useful. The web site tabs to refer to previously visited web sites is very useful. In the future there should be an intelligent link between web page (via tabs) an content in Work pad. If the web site is open then the text highlights in the work pad. Alternatively, if content is clicked in the work pad the user can access the relevant section in the web site it is copied from.

Participant 3 thought that the option of using two browser windows was the best thing about the tool. The work pad functionality is also useful.

Participant 4 believes that using tabs to look back at previous web sites is wonderful. It can give quick comparison even if the content will not be recorded. The name Work Pad is a very good one. Once the text is selected for copying into work pad, the text does not de-select before pasting into Work Pad.

Participant 5 thought that recording web content on the work pad and typing in the source of the information is the best feature of the tool. It is also useful that users can refer back to previously visited web sites. However, the combo box entries were more helpful than the tabs. The participant didn't use the tabs, however he/she feels that competent web users may find it handy mechanism. The option of having split screens as well as maximizing the work pad or browser window is also designed very well.

Question 3

What was the worst thing about the tool? What did you like least about the tool?

Participant 1 felt that there should a search list of web sites that users can go back and visit. The work pad and browser window could not be resized.

Participant 2 did not like having to hide and show the work pad, he/she would much rather have the work pad displayed on the interface all the time, but be able to resize it so that a few lines can be seen of the work pad window.

Participant 3 felt that the interface is not user friendly. It could be more similar to the Internet Explorer browser, with similar toolbars, presentation, bookmarks and back and forward buttons. The full URL should be shown rather than just the upper hierarchy.

Participant 4 felt that there should be a back button, windows should be resized. There is no need for a duplicate close button for the tool. Work Pad could have distinguished font, the lines could have single spacing and users should be able to change font later if required.

Participant 5 thought that the tool needs more color with fancy font. The tool does not have bookmarks or back and forward buttons. Each buttons could have hints or descriptions of the purpose of the button. He felt that users should be able to write in words in the URL box as well as specific URL locations.

C.3 Additional Comments

Participant 1
Hard to grasp and apply
Start from scratch with searches no back
Tabs reloaded websites and not save the last level visited
Can not close tabs
Close browser button and a close tab button
Tab system is not user friendly
Preferred vertical
Did not understand show/hide workpad
Did not understand vertical/horizontal window icons
Did not find dragging down easy
Wanted right click menu copy paste option
Search list of last sites
Back button
Resize windows option
Spent too long collecting web sites could not reach compare stage
Found 2 browser window arrangement good
Close application button is redundant
Buttons are too large
I thought that the cross by the browser window was for selecting and closing the tabs
sometimes tabs do not display related sites or any sites
If a user types a url in error, it should not create a tab
The horizontal layout with workpad below the browser, makes copied text hard to read
Not obvious that to resize a browser the workpad needs to be hidden
Hidden work pad seems to be part of browser section, could minimise workpad
Should have option that all new websites can open in one window and overwrite the tab or as a list of tabs

Participant 2
Can't resize browsers would be handy
Status- progress bar
Minimise windows not hide/show
Memory to save web pages in cache not reload
No overwrite when copy paste
Open new windows in IE, tool is front end to IE
Horizontal layout
Frequent workpad, difficult to find workpad in amongst layers windows, select from Browser find pad then place to drag
Visited tabs frequently accessed, restart and retype
Combo box last entry appears not recent, same as tabs
Compare 3 for last two tasks b, c, and minimal time, easy to learn
Tabs can be resized
Option to close window and tabs
URL, site overwrites existing window - a line of tabs or option to show all tabs in one window a time option

Hints show whole name of tabs
Hide/show defeats the purpose should have work pad displayed (small view) or minimized at all times

Participant 3
Where are a browser standard toolbars?
I don't see links or bookmarks?
If I select a whole page of text it does not copy onto workpad
I'll do copy and paste the old fashioned away and write onto paper
there is no back button
When I'm comparing two browser windows the work space is crammed so once I finish I'll close a browser
I have to log in each time?
Cursor is not in position
When I open a new window in IE I need to use Ctl C and Ctl v to copy into workpad, window switching
Cannot resize each window
Pressed enter to go back to visited sites
First round did not access workpad , but then learnt to use and used it frequently
Preferred horizontal layout with one window or two in the first task
Needs to make it look more attractive
Should show full URL rather than just the upper hierarchy
Not quite user friendly
Liked two browser windows and work pad

Participant 4
Why duplicate close button in top right hand corner?
Resize windows would be a helpful option
No back button
Horizontal and vertical can show with two boxes line vertical line in between
Two boxes horizontal line in between
Like the tabs- going back to site previously viewed
The name 'work pad' is a good one
Copying and selected the content does not deselect, before pasting into work pad
Prefer single spacing in work pad
I had trouble finding hide/show toggle button to resize the browser
Can't copy banners or graphics onto work pad
Need to press enter in the URL box to go back levels as there is no back button

Participant 5
Can't just write words in URL box have to type in full url location
No back button
Reloading full site to first level
Go back to first tab the screen is clear
Pressed close browser button instead of hide workpad to resize browser window
No individual resizing
Must press 'search' button
No status

type in url shows past site, then reloads the recent site
New windows open in Internet Explorer, I minimized this window and placed next to work pad so that it is easier to drag text across
Should not need to switch between IE browser and Networker can get confusing
User input and dragging onto workpad I liked
Useful to go back to previous sites
Liked split screens and option to maximize
Didn't use tabs but competent users may require them
Needs more color
Fancy font and lettering
No bookmarks to refer to favorites list
Buttons could have hints
Liked vertical split screens it meant I can read like a text book, read down from left to right.

C.4 Participant Performance

Table C.2 Task A

	Time (min)	Success/ Failure	Browser 1	Browser 2	Horizontal Layout	Vertical Layout	Workpad accessed	Visited web site	Comparison #	Name
Participant #										
1	10+	F	1	1	0	1	1	1	0	-
2	10+	F	1	0	1	1	1	2	0	amazon
3	9.35	S	1	1	2	0	2	3 (enter)	3	yellow pages amazon b +n
4	6.37	S	1	1	0	1	1	1 (enter)	3	Amazon Yahoo B + N
5	9.10	S	1	0	0	1	3	2 (enter)	3	Yahoo Amazon B + N

Table C.3 Task B

	time (min)	Success/ Failure	Browser 1	Browser 2	Horizontal Layout	Vertical Layout	Workpad accessed	Visited web site	Comparison #	Name
Participant #										
1	9.45	S	1	0	1	1	1	1	1	all bank rates in table
2	7.08	S	1	0	1	0	1	1	3	National BNZ ANZ
3	5.23	S	1	0	3	1	2	0	2	National BNZ ANZ
4	3.57	S	1	0	0	1	2	0	1	National BNZ ANZ
5	5.40	S	1	0	1	0	4	1 (enter)	4	BNZ ANZ Kiwibank National

Table C.4 Task C

	time (min)	Success/ Failure	Browser 1	Browser 2	Horizontal Layout	Vertical Layout	Workpad accessed	Visited web site	Comparisons #	Name
Participant #										
1	10+	F	1	2	0	1	1	0	0	-
2	9.05	S	1	0	0	0	3	3	3	alglen apollo aotea
3	4.39	S	1		3	1	2	0	3	alglen apollo aotea
4	6.05	S	1	0	0	1	1	3 (enter)	3	aotea alglen northlands
5	6.53	S	1	0	0	1	3	0	3	apollo aotea alglen