A workbench to support development and maintenance of world-wide web

Susannah Dalton

A Workbench to Support Development and Maintenance of World-Wide Web Documents

Susannah Dalton

M.Sc. Thesis

Centre for Software Maintenance

Department of Computer Science

University of Durham

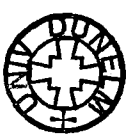
1996

ABSTRACT

The World-Wide Web is one of the most dominant features of the Internet. In its short life it has become an important part of information technology, having a role to play in all sectors. Unfortunately, it has many problems too. Due to its fast evolution, World-Wide Web document development is undisciplined and has resulted in the appearance of much poor quality work. This is also widely due to the inexperience of authors, the lack of conventions, standards or guidelines and useful tools for development and maintenance of Web documents.

One solution to the major problems of poor quality of World-Wide Web documents is the improved maintenance of such documents. Maintenance is an important area that, similar to software engineering, receives little attention compared with development.

In order to address the problems of World-Wide Web document maintenance, research into the area was carried out through a literature survey and case studies of the organisations that manage World-Wide Web sites. The results of this research led to producing a workbench which provides support to both developers and maintainers of Web documents. This workbench consists of methods, guidelines and tools for World-Wide Web development and maintenance.



Declaration

No part of the material offered has previously been submitted by the author for a degree in the University of Durham or in any other University. All the work presented here is the sole work of the author and no one else.

CONTENTS

CHAPTER 1 12

INTRODUCTION 12

1.1 Objectives of the Chapter 12

1.2 Introduction 12

1.3 Research Area and Criteria for Success 13

1.4 Outline of rest of thesis 14

1.5 Summary of Main Points 14

CHAPTER 2 16

HYPERTEXT AND WORLD-WIDE WEB MAINTENANCE 16

2.1 Objectives of the Chapter 16

2.2 Introduction to Hypertext 16

2.2.1 Definition of Hypertext 16

2.2.2 History of Hypertext 18

2.2.3 Significance of Hypertext 18

2.3 World-Wide Web 19

2.3.1 The Origins of the World-Wide Web 19

2.3.2 Defining the World Wide Web 20

2.3.3 How does the Web work? 20

2.3.4 The Advantages 21

2.3.5 The Problems 21

2.3.6 Future of the Web 23

3.1 Objectives of the Chapter 48

3.2 Case Studies 48

3.3 Discussion on the CCTA Web Site and Management Procedures 48

3.3.1 Good Points 49

3.3.2 Problems 50

Recognised Problems 50

Other Problems 50

3.3.3 Points for Consideration 51

3.3.4 Suggested Improvements 52

3.3.5 Foreseeable Problems 53

3.3.7 A Summary of the CCTA Government Information Service 54

3.4 IT Service Case Study 54

3.5 Discussion on the ITS Web Site and Management Procedures 54

3.5.1 Good Points 54

3.5.2 Problems 55

3.5.3 Points for Consideration 56

3.5.4 Suggested Improvements 57

3.5.5 Foreseeable Problems 57

3.5.6 A Summary of the IT Service 58

3.6 Conclusions from Case Studies and Major Problems Identified 58

3.7 Developing a Solution 63

3.8 Summary 63

CHAPTER 4 64

DEVELOPMENT OF A SOLUTION 64

4.1 Objectives of the Chapter 64

4.2 A Proposed Solution 64

4.3 Requirements analysis 64

4.3.1 CCTA Requirements 65

4.3.1.1 DOS-UNIX Problems 65

4.3.1.2 Managerial Problems 65

4.3.1.3 Technical Problems 66
4.3.1.4 Development 66
4.3.1.5 Maintenance Problems 66

4.3.2 ITS Requirements 67
4.3.2.1 Management 67
4.3.2.2 Development 67
4.3.2.3 Maintenance 67

4.3.3 Own Requirements 68
4.3.3.1 Development 68
4.3.3.2 Maintenance 68

4.4 Synthesis of Requirements 69
4.4.1 Requirements for Management 69
4.4.2 Requirements for Development 71
4.4.3 Requirements for Maintenance 71

4.5 Requirements Specification 72

4.5.1 Introduction 72
4.5.1.1 Purpose 73
4.5.1.2 Scope 73
4.5.1.3 Overview 73

4.5.2 General Description 73

4.5.2.1 Development 73
4.5.2.1.1 Content 74
4.5.2.1.2 Navigation 74
4.5.2.1.3 Structure 74
4.5.2.1.4 Learning HTML/Java/VRML 74
4.5.2.1.5 Developing for maintenance 75

4.5.2.2 Maintenance 75
4.5.2.2.1 Structure Maintenance 75
4.5.2.2.2 Quality Checking 75
4.5.2.2.3 Maintenance Documentation 76
4.5.2.2.4 Version Control 76
4.5.2.2.5 Java Maintenance Problems 76
4.5.2.2.6 Removing Pages 77

4.5.2.2.7 Navigation and Visualisation 77

4.5.2.5 Management **77**

4.6 Design **78**

4.7 Implementation **82**

4.8 Summary **84**

CHAPTER 5 **85**

RESULTS AND TRIAL USAGE OF WORKBENCH **85**

5.1 Objectives of the Chapter **85**

5.2 Does the Workbench Conform to the Original Requirements? **85**

5.2.1 Requirements for Management 85

5.2.2 Requirements for Development 86

5.2.3 Requirements for Maintenance 86

5.2.4 Result 86

5.3 Trial Usage **87**

5.3.1 Development Section of Workbench 87

5.3.2 Maintenance Section of the Workbench 88

5.3.3 Management Section of the Workbench 88

5.3.4 Future Developments of the Workbench 89

5.3.5 CCTA Trial Usage 89

5.4 Overall Assessment of the Workbench **89**

5.5 Summary **89**

CHAPTER 6 **91**

CONCLUSIONS **91**

6.1 Objectives of the Chapter **91**

6.2 Synopsis of Work **91**

6.3 Criteria for Success 92

6.4 Appraisal 93

6.5 Future Work on the Workbench 93

6.6 Future Work in this Area 93

6.7 Summary 94

APPENDIX A 95

TOOL LOCATION 95

Tool Name 95

Tool Location 95

HTML Validators 95

Link Validators 95

Server Log File Analysers 95

APPENDIX B 96

CCTA CASE STUDY 96

B. General Overview of Objectives of the CCTA with regard to the World Wide Web 96

B.1 CCTA Government Information Service 96

B.2 The Team Structure 96

B.3 The Structure of the Web Site 97

B.3.1 System put in place for the set of CCTA pages 97

B.3.2 Development of Web Site in the Future 100

3.2.3.6 The Tool 100

APPENDIX C 103

IT SERVICE CASE STUDY 103

C.1 Information Technology Service 103

C.2 The Personnel 103

C.3 The Structure of the Web Site 103
C.3.1 System put in place for the set of Durham Web pages 103
Regulations 104
Rules of Presentation of Official University Pages 105
Guidelines for Presentation 105
C.4 Development of the Web Site in the Future 107
C.5 Tools 107

ACKNOWLEDGEMENTS **108**

REFERENCES **109**

LIST OF FIGURES

Figure 1 The Lynx Browser 26
Figure 2 The Hotjava Browser 27
Figure 3 The Netscape Browser 28
Figure 4 An example of a possible background 30
Figure 5 A browser that supports tables 60
Figure 6 A browser that does not support tables 61
Figure 7 An indexed guided tour 79
Figure 8 The structure of the development section 80
Figure 9 The structure of the maintenance section 81
Figure 10 The structure of the management section 82
Figure 11 A screenshot of the workbench 83
Figure 12 CCTA management model 98

CHAPTER 1

INTRODUCTION

1.1 Objectives of the Chapter

There are two main objectives of this chapter. The first objective is to describe the field of work and the main line of research within that field. Following on from this, the criteria which will make this work successful will be defined. The second objective is to describe the outline of the rest of the thesis.

1.2 Introduction

In the world of information technology, the Internet has become one of the most dominant features. The Internet is the term used to describe the world-wide network of computer networks. The idea was developed by the US Department of Defence in the late 1970s. This was known as the ARPANET, which was an experimental wide area network. During the 1980s it increased in popularity, first with governmental agencies, then academic institutions, followed by private research laboratories and finally corporations and individuals began to interconnect their computers, thus making the Internet cover the globe, becoming of major importance [1]. It allowed quick and efficient communication through electronic mail and facilitated information sharing with a host of on-line databases and archives, hence its popularity. However, it became apparent that the needs of the users of the Internet were greater than what was provided by the facilities available. Problems developed from users attempting to share information using different platforms and different software. This soon imposed limitations on the information sharing process which led to the birth of the World-Wide Web. The World-Wide Web is an Internet information service, with an easy to use graphical display which allows information retrieval via textual relations. It is also known as the Web, the WWW or W3.

The two key features of the World-Wide Web are platform independence and the use of hypertext to present information. Hypertext is non-sequential writing with a database method and a presentation scheme. In this thesis we will focus on this second feature, that of hypertext.

Hypertext presents an easy-to-use interface which makes it popular. It also allows great freedom in the presentation of the information. The language required to create World-Wide Web applications is Hypertext Mark-up Language, more commonly known as HTML. HTML is simple to use, not requiring a computer science background to master it. And so, the World-Wide Web is accessible to the masses, contributing to its popularity.

1.3 Research Area and Criteria for Success

The work for this thesis is centred around the subject of hypertext and the World-Wide Web. It is being carried out at the Centre for Software Maintenance, Department of Computer Science, University of Durham.

The World-Wide Web has developed in its short life very rapidly. It came into existence only in 1992 and is now the main focus of the Internet. As with software engineering, its growth has been rapid and undisciplined. This unfortunately meant that the appearance of poor quality documents with broken links in the structure, poorly designed interfaces, badly written English (or other language) and incorrect information among other problems have arisen.

The area of research for this thesis is the maintenance of hypertext. The research is centred round hyperdocuments, also known as hypertext applications. Hyperdocuments are a set of World-Wide Web pages which constitute a whole document, often simply referred to as a document.

Given the nature of hypertext, it is very easy to write a hyperdocument and there is much freedom on the structure of the document, allowing a variety of representations of information. The target of this research is to provide an overall approach to the maintenance of these documents and the development of hyperdocuments so that they are more maintainable. In order for this research to be considered a success, the following criteria need to be achieved:

- An investigation into the problems that exist in this area.
- An approach to overall development and maintenance of hyperdocuments that helps to solve some of the problems identified in the investigation.

1.4 Outline of rest of thesis

This section will outline the remainder of this thesis by describing the content of the following chapters.

Chapter two will be an introduction to hypertext and the World-Wide Web. It will define hypertext application maintenance, current practices in World-Wide Web hypertext application maintenance, it will explain the need for maintenance and the problems that currently exist. It will then go on to discuss the previous work done in this area, including analysing hypertext applications and surveys existing tools for maintenance, visualisation and navigation.

Chapter three is in two parts. It describes two case studies, that of the CCTA and the Durham University IT Service. These case studies describe the approach of both bodies to maintenance and management of a Web site and attempts to establish good points and problems with the individual organisation's approaches as well as assessing their needs. The chapter then goes on to give a summary of the problems and the criteria for a successful solution.

Chapter four describes the proposed solution to the problem and gives the full requirements analysis and specification referring to the case studies and the author's own experience.

Chapter five gives the results and trial usage of the tool. This is done by checking the results against the original requirements and the original problem area defined. A trial usage is performed to see if the solution proves useful and appropriate. In conclusion of the chapter, an overall evaluation of the solution is given.

Chapter six evaluates the results of the thesis and discusses whether it meets its objectives. This is done by discussing the results achieved, whether they address the original problem and whether they solve the problem conclusively. In the case where problems still exist, what work would further the work done in the thesis is described. The overall conclusion is given in chapter seven.

1.5 Summary of Main Points

The World-Wide Web came into being to fulfil the needs of the users of the Internet. It rapidly became popular, due to its ease of use and its two key features, platform independence and hypertext. Due to its popularity the World-Wide Web has undergone a rapid growth and

problems have arisen, particularly in the area of development and maintenance of World-Wide Web applications.

The research area for this thesis is to investigate the main problems in the area of hypertext maintenance and from there develop an approach to overall development and maintenance of hypertext applications which will help solve some of the problems identified.

The thesis will consist of seven chapters, containing a literature survey, two case studies, a discussion of the problems, a devised solution, an evaluation of the solution and a conclusion to the work. The following chapter will talk in more depth about both hypertext and the World-Wide Web. It will define hypertext maintenance. It will look at the previous work done and will identify the current problems.

CHAPTER 2

HYPERTEXT AND WORLD-WIDE WEB MAINTENANCE

2.1 Objectives of the Chapter

In this chapter, hypertext is defined, giving a history and describing its significance. The World-Wide Web is defined. Its origins are discussed and a description of how it works is given. Some of the advantages of the Web are then outlined, followed by a discussion of the problems and its future. The chapter goes on to set out what World-Wide Web maintenance is perceived to be, why it is needed and the problems that exist with it. This is followed by a survey of previous work done in the area of World-Wide Web maintenance. This work includes work done with standards and guidelines, analysing hypertext and World-Wide Web documents, surveying existing tools and work with visualisation and navigation of the World-Wide Web. In summary, the chapter looks at the research area in more depth and identifies problems in each section drawing from previous work done in the area.

2.2 Introduction to Hypertext

The main topic of this thesis revolves around the subject of hypertext. This section gives an introduction to the subject of hypertext and how it came about, culminating in a description of the significance of the concept of hypertext and why it is so important to information systems.

2.2.1 Definition of Hypertext

Ted Nelson who coined the phrase "hypertext", defined it as:

> "a combination of natural language text with the computer's capacity for interactive branching or dynamic display...of a

> non-linear text...which cannot be printed conveniently on conventional paper".

Hypertext is essentially non-sequential writing with a presentation scheme and a database method. However, one should be aware that hypertext is not hypertext if there is simply an underlying database, for example windows or file systems, to qualify as hypertext, it must use the more sophisticated notion of links, which allow active cross-referencing [2]. Links are one of the two building blocks of hypertext, the other being nodes.

Links are similar to pointers, they provide connections between one document and another or from one part of a document to another part of a document. These links fall into two categories, referential and organisational [2]. Referential links provide references, some examples would be links between document references to documents, keywords to text and entries in tables to longer descriptions. Organisational links implement hierarchical information i.e. links between successive documents. It is these links that provide the non-linear organisation of text, which is the key feature of hypertext.

The concept of hypertext leads to the more general concept of hypermedia, which involves not only text, but graphics, sound, animation, video clips and interactive applications.

Nodes are items which contain any form of media such as text, graphics, animation or video clips. Nodes can contain many links to other nodes, these are indicated by "anchors", which could be for example, a piece of highlighted text. This text should be suggestive of the content of the destination node.

Ideally, hypertext consists of the following features:

- the database is a network of textual or graphical nodes which can be thought of as a hypertext application;
- windows on screen correspond to nodes;
- standard window systems operations are supported;
- windows can contain any number of link icons (link icons should be suggestive of the content of the node);
- users can create new nodes and links;

- the database can be easily browsed: by following links, by searching networks for a string, navigating around a hypertext application using a browser.

2.2.2 History of Hypertext

The history of hypertext is rich and varied because the concept behind hypertext of cross-referencing is not a new idea [2]. Cross-referencing is a very old concept, dating back to the written documents which is as far back as 3000BC. Cross-referencing that appears in books is known as manual hypertext.

Hypertext systems were developed for macro literary systems, which is the study of technologies to support large on-line libraries in which inter-document links are supported. They were also developed for problem exploration tools, which are tools to support early thinking on a problem when many disconnected ideas come to mind. Hypertext systems are also useful for browsing systems and general hypertext technologies which are general purpose systems designed to allow experimentation with a wide range of hypertext applications [2].

2.2.3 Significance of Hypertext

Before the advent of hypertext, most modern computer documents were linear and linear organisation was considered adequate for the documentation of computers [2]. With the idea of using the power of a computer to allow almost instantaneous access to references came the idea of hypertext. This opened a whole new way of presenting information. It made the presentation of information far more flexible and less tied to the one linear structure. At the time of Nelson it became known as the basis for global literature [2] and later it was to become the basis of the World-Wide Web.

Why use hypermedia? It is the science of relationships. It provides direct access to information, it has navigation, annotation and information overviews which enhance applications. It allows collaborative authoring, document management, customisation, education and corporate training [3] [2]. It provides users with access to the contents and connections of content. However, hypertext does have its problems. The main problem, which was already recognised in 1987 [2] was that of disorientation. It also provides a greater cognitive overhead to the reader as they must continually work out how to get to their destination, as hypertext presents no natural topology.

2.3 World-Wide Web

This section is an introduction to the World-Wide Web. It contains a section on the origins of the Web and a definition of the Web. It then goes on to discuss the advantages and disadvantages and the future of the World-Wide Web.

2.3.1 The Origins of the World-Wide Web

The idea behind the World-Wide Web was conceived by Tim Berners-Lee at CERN [4]. He saw that scientists in his field of high energy physics, had difficulty collaborating together due to problems accessing and sharing information. These problems came about as a result of scientists using a range of different protocols and workstation types and variable display facilities.

This brought about the development of the World-Wide Web. The intention was to facilitate the information sharing on all aspects of a non-trivial project [4], [5].This idea of a Web was prompted by a positive experience of a home-brew personal hypertext system used for keeping track of personal information on a distributed project [6].

This experience led to the idea of providing a more sophisticated and user-friendly Internet information service. The intention of the designers was to provide an easy to use piece of software which operated in a consistent manner [7]. They decided on a hypertext system, which meant a simple 'point and click' interface. The Web was designed so that information change would smoothly reshape to represent the new state of knowledge. It was able to expand rapidly without any problems, revealing it knew no boundaries of knowledge or geography [6].

Later the World-Wide Web became a test bed for various sophisticated hypermedia and information retrieval concepts [8]. It quickly became more sophisticated and its usage began to spread, it was then adopted by other fields other than high energy physics [8].

Currently the World-Wide Web is developing at a fast rate. It is used by academic institutions, government agencies and commercial organisations, as well as individuals who use it purely for fun. It is becoming more sophisticated with the introduction of Java, which is a programming language that enables users to create interactive pages. The Virtual Reality Modelling Language (VRML) is also widely used, which permits some virtual reality extensions.

across pages. Images can be included within documents, and forms with input boxes and radio buttons can be generated.

How these documents are viewed depends on the browser being used. A simple text-based browser may simply display text in paragraphs and lists, whereas a more complex browser may allow the inclusion of a range of images and the ability to pass files they cannot decode to external programs, thus if the browser does not have the facility to play movies, it can be configured to pass the movie file to a separate movie player on the computer and use that to display movie images.

2.3.4 The Advantages

The World-Wide Web has many advantages which is the reason it has become so popular. It is by far the easiest information system ever provided by the Internet. Information is simple to browse, requiring no real computing background knowledge. Due to its popularity and the large number of users, it has become an important information source which is useful to academic research and commercial activities such as advertising and so forth. The World-Wide Web has provided the possibility of integrating Internet services and has become a test bed for new document technology.

2.3.5 The Problems

The Web with its success ran into various problems. Its popularity led to a larger number of people using the Web. The increased numbers, raised the demand which lead to a quick evolution. In consequence of this there has been a lack of tools and poor quality of documents has resulted. All the while the Web has remained a complex system with a number of constraints and the problems that had existed previously with hypertext systems are still problems which still remain to be solved.

The Web's evolution could be described as an avalanche of servers, documents and hyperlinks compounded by an exponential growth in Web usage. The growth of the Web has been caused by a continuing pressure to expand through the demands of the user community. It is the belief of some that this has all but buried its usefulness to the real world [11]. Its fast evolution has certainly contributed to some of its major flaws.

In the first instance this has meant that the Web does not meet its design goals, as being an easy to use and up-to-date pool of knowledge [8]. Use of the Web is still reasonably complex.

HTML is not simple to use and there are few sophisticated editors. It is difficult to configure and maintain a server so that it remains reliable and presents few security risks.

The WWW was used as a test bed for various sophisticated hypermedia experiments and information retrieval concepts. These concepts were quickly adopted by the general Web community which means that established parts of the Web are only experimental [8].

A fast evolution has also meant that constraints exist. HTML can be quite restrictive. Users cannot, for example, specify the font in which their document is displayed. Many users require more sophisticated data formats. HTTP does not support communication between servers which has implications for data integrity [8].

Determining quality can be difficult. Most users use as an approach to quality control in areas such as software engineering, the aid of standards and guidelines available or even the general consensus of opinion to guide them. Unfortunately as yet, there are no such standards or guidelines and a very conflicting consensus on what a good hyperdocument is, which eventually leads to poor quality [12].

The original problems that exist with hypertext have not yet been resolved. The most notable being defined by [2] as the crucial problem: user disorientation during navigation, this problem according to [12] still hasn't been resolved satisfactorily.

With this fast evolution comes a lack of adequate methods and tools for development and maintenance [3, 12]. Despite there being much Web development software, most software tools available are still fairly limited in their capabilities. Software tools for development rarely hide the features of HTML and thus require a good knowledge of how to develop Web documents. There is limited help in these tools on producing a design for a hypertext application. There are also few tools that provide proper testing, debugging and evaluation features. This means that poorly written hyperdocuments cannot be thoroughly tested and maintained. Better tools are also required to analyse hypertext structures and documents properly. Tools are also required to judge the maintainability and readability [12]. There is no sophisticated support for collaborative authoring. Virtual reality extensions are required and finally tools for all platforms are needed not just Windows.

This lack of standards and tools has led to problems of authorship quality, where authors have little experience[13, 12]. Due to the simplicity of browsing, there is a tendency to believe that authoring will be simple too. Inexperience has also contributed to problems of social code. The Web is easy to use and used by a larger number of people who have not previously used the Internet information services and are unaware of its social code. Many authors are relatively unused to using the Internet and do not observe etiquette of multi-user systems, such as News. One of the problems with WWW architecture is that it has not been subjected to the rigours of formal development [8].

It is thought that the WWW development community are largely ignoring the problems of authoring [8, 12, 9]. Collaborative use of information creates such problems as lost updates, false cross-references and concurrency problems with database design. WWW does not provide database management systems' restrictions to solve these problems. HTTP only really provides support for collaborative reading, not creation, deletion and modification.

2.3.6 Future of the Web

If the Web is to improve, it should contain standards, guidelines and conventions. It requires better tools, fewer restrictions than currently exist with HTML and HTTP and an evolution which incorporates new technologies.

It has been established by the Web community that there is a lack of good development and maintenance tools for use with the World-Wide Web. Web development tools need to become more sophisticated allowing non-expert users to manipulate hypertext efficiently. Tools for designing Web pages need to be developed, as well as tools for maintenance and thorough testing. Tools which support collaborative authoring would increase the usefulness of the World-Wide Web.

HTML is widely regarded as being restrictive and thus requires development so that it is less so, allowing for example choice of fonts by the author and more sophisticated document type definitions providing for the needs of commercial publishers of on-line material amongst other things.

Setting up and maintaining a Web site is complex, in order to facilitate this, easy-to-use servers for low end machines to make ease of publications for small groups and individuals would be useful.

On the more speculative front, the following ideas would make the Web even more powerful and of higher impact: the development of a common format for hypertext links for two and three-dimensional images; integration with concurrent editors and other real-time features such as teleconferencing and virtual reality; evolution of objects to contain more machine-oriented semantic information allowing more sophisticated information, as opposed to human-readable only documents and improved navigation through new technologies.

2.4 World-Wide Web Maintenance

This section gives a description of what World-Wide Web maintenance is, why there is a need for it and what the limitations are with current maintenance practices.

2.4.1 Definition of World-Wide Web Maintenance

There is no standard definition for World-Wide Web maintenance. However, maintaining WWW pages is generally perceived to be making changes. In the world of software engineering, software maintenance is defined to be:

> "the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a changed environment." [14, 1983]

At the Centre for Software Maintenance, the definition is modified to include the concept of preventive maintenance:

> "Software maintenance is the set of activities both technical and managerial, that ensures that software continues to meet organisational and business objectives in a cost-effective way."

To make a more precise definition of World-Wide Web maintenance, I have looked to software maintenance. To use a similar definition to the first, would mean introducing the concept of a WWW hypertext application development lifecycle, where the document would reach a point of delivery. This is difficult to define for WWW pages. The simple appearance of a hypertext application on the Web would seem a logical point of delivery. However, pages are often put on the Web with signs attached to them to indicate that they are under construction. To try and

clarify at what point delivery is made, it would seem that if a page is on the WWW and is not visibly under construction, i.e. it does not indicate anywhere that the page is under construction, it is said to be delivered.

At this point, changes that are made to the pages would be maintenance. However, these changes, like in the definition of software maintenance, should be changes which constitute an improvement. These changes should correct faults, improve the "performance" of the page, "adapt" the pages or "perfect" the pages to changing requirements. "Improving performance" would mean improving the way in which a page achieves its goal. Adaptive maintenance is the set of activities that take place as a response to a change in the environment, for example, to adapt software to an upgraded operating system or a different platform. This seems less relevant to Web maintenance as one of the features of the Web is platform-independence. Hence, most hardware difficulties are already overcome. However, Web pages are viewed differently on different browsers. In *Figure 1 The Lynx Browser* is shown, in *Figure 2 The Hotjava Browser* is shown and in *Figure 3 The Netscape Browser* is shown. These browsers all show the same page, this gives an idea of the possible discrepancies that arise between browsers. In a sense the browser could be viewed as a type of operating system on which the users can operate their documents. It should be possible to view a document on any browser and achieve the same goal. If a new browser is brought in or new features are added to a browser, changes could be made to accommodate the browser or these new features. As with software maintenance, there are often changes in the requirements of the pages, which would entail perfective maintenance of the pages.

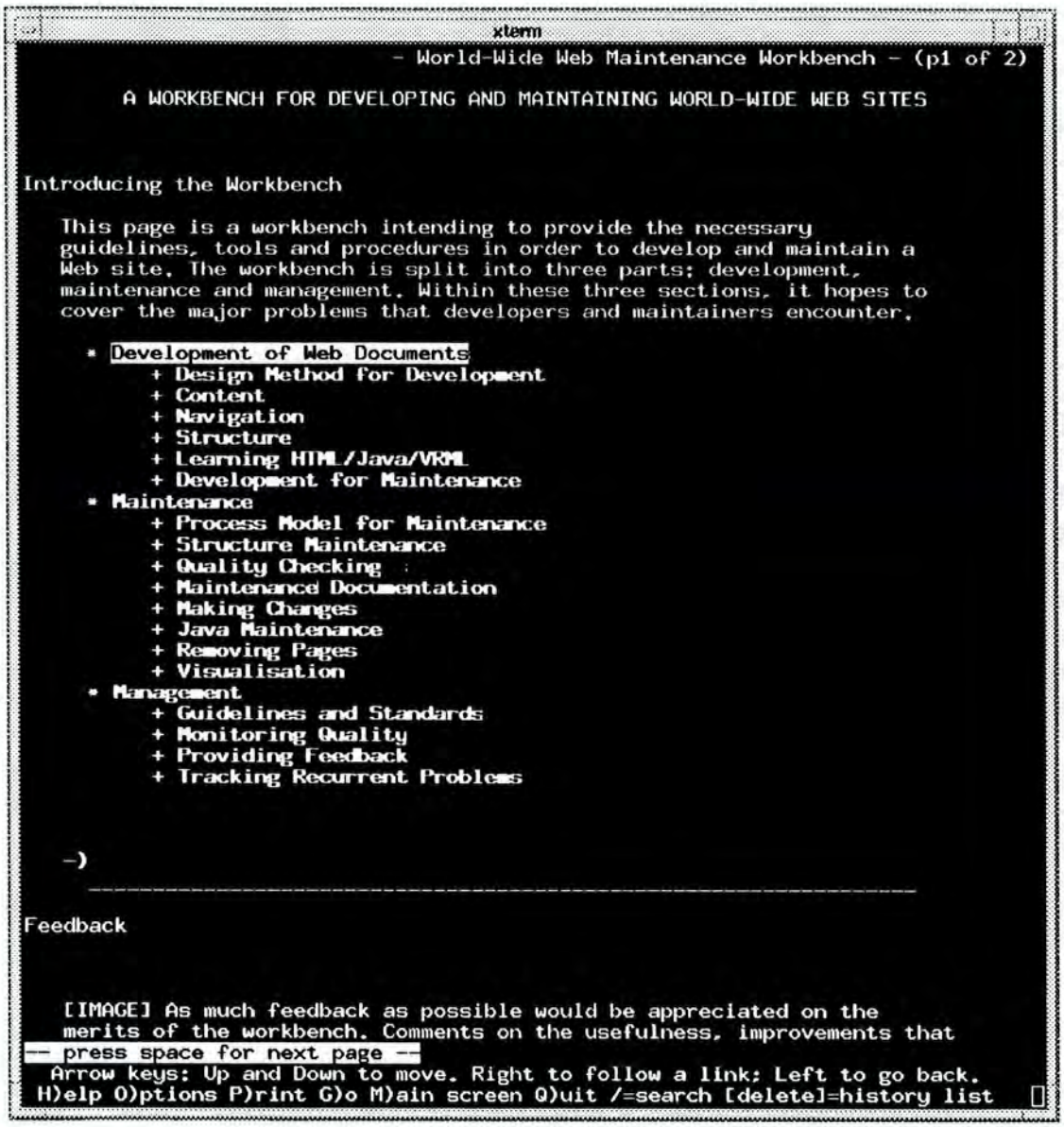


Figure 1 The Lynx Browser

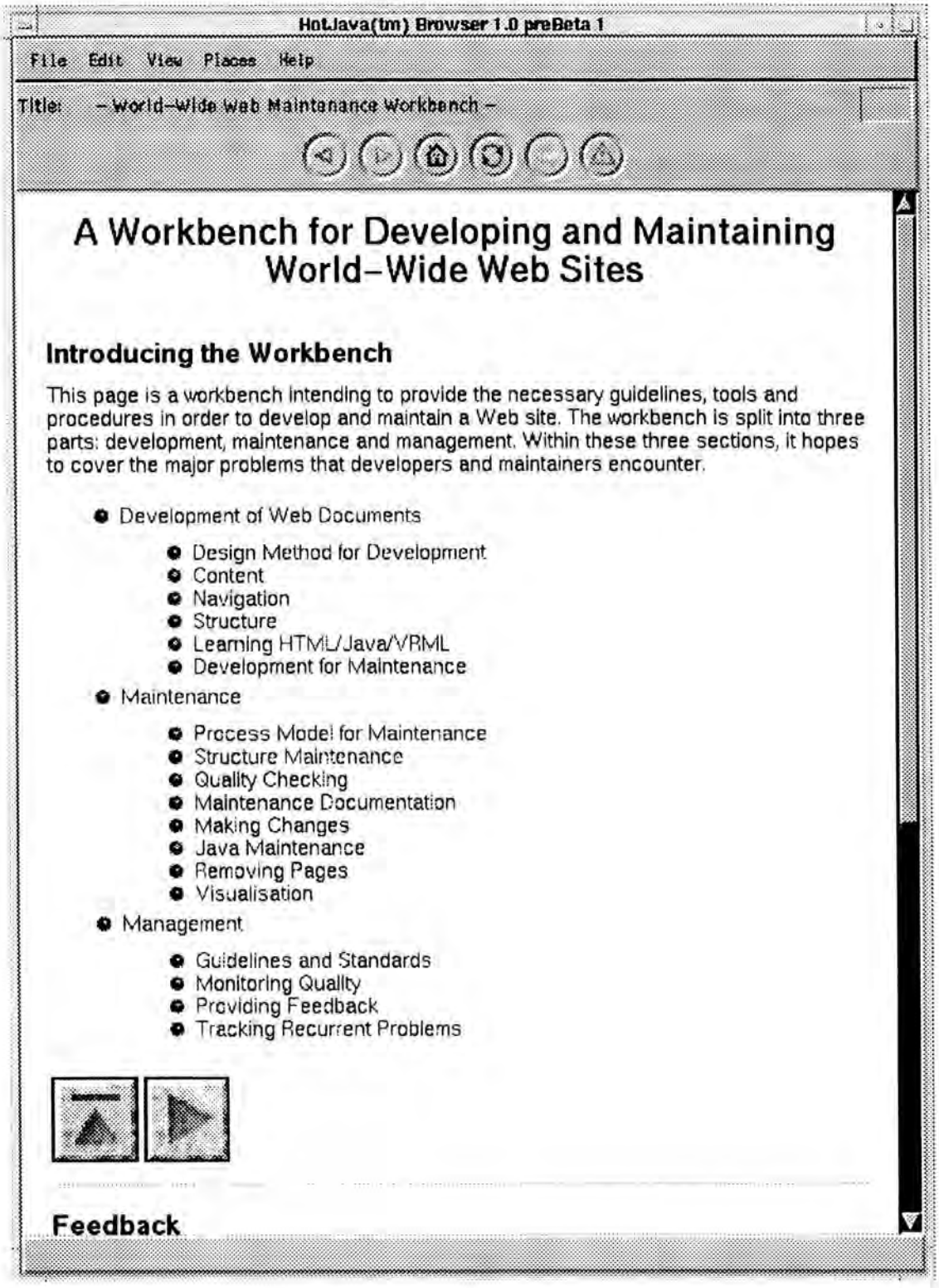


Figure 2 The Hotjava Browser

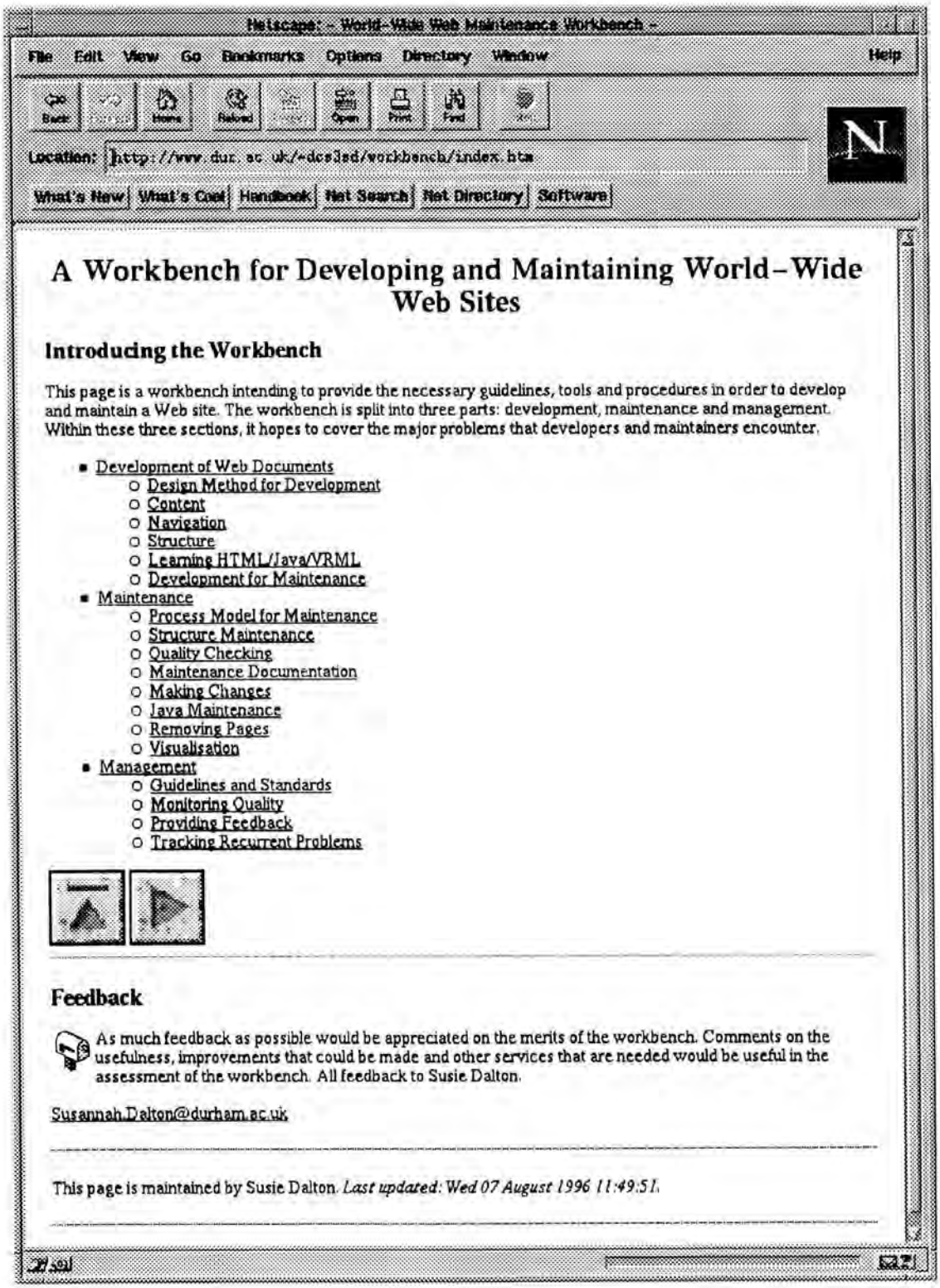


Figure 3 The Netscape Browser

If the second definition of software maintenance defined by the Centre for Software Maintenance, is taken into account, then the concept of preventive maintenance needs to be introduced. Preventive maintenance is described as the set of activities which take place in

order to prevent malfunctions and improve maintainability. Unlike other maintenance activities, preventive maintenance is not usually undertaken as a direct response to a problem or a requested change, but is identified as a need from within the maintenance team [15]. A similar type of maintenance is also needed for Web maintenance. An example of preventive Web maintenance would be providing a good structure to the user's document to make it easier to add to and delete from.

Although this thesis concentrates on maintaining Web documents, it is important to mention Web sites too. One could consider a Web site as a superset of Web hyperdocuments. However, it would be wrong to assume that all the activities that apply to Web hypertext application maintenance automatically apply to Web site maintenance. If the homepage of a site is considered to be an individual document then site maintenance involves monitoring the structure of the site, the security, standards and guidelines within that site, so there is much in common between these two activities.

2.4.2 The Need for World-Wide Web Maintenance

One of the reasons for storing data on-line is that it makes it easy to change. In the case of the Web, the information stored on there has the attraction of being dynamic. Most applications designed for the World-Wide Web contain volatile data that requires frequent updating. If the information is going to evolve dynamically without major problems, a maintenance strategy is required [16, 17].

Web maintenance is not simply about updating the content of the document. There are many more aspects to it. Maintenance divides into three issues broadly speaking, structure maintenance, quality of content checking and preventive maintenance.

Maintaining the structure means ensuring that links, graphics, ftp, file and mailto facilities all work correctly. Maintenance of the quality of content involves ensuring that the information that is presented, is well presented with correct spelling and grammar, that it can be viewed on all browsers, that it does not involve over elaborate features such as a gaudy background (see *Figure 4 An example of a possible background*) or flashing text, that the content is correct and that the pages achieve their goal. Preventive maintenance is the act of making the user's pages more maintainable. This for example is creating well structured pages.

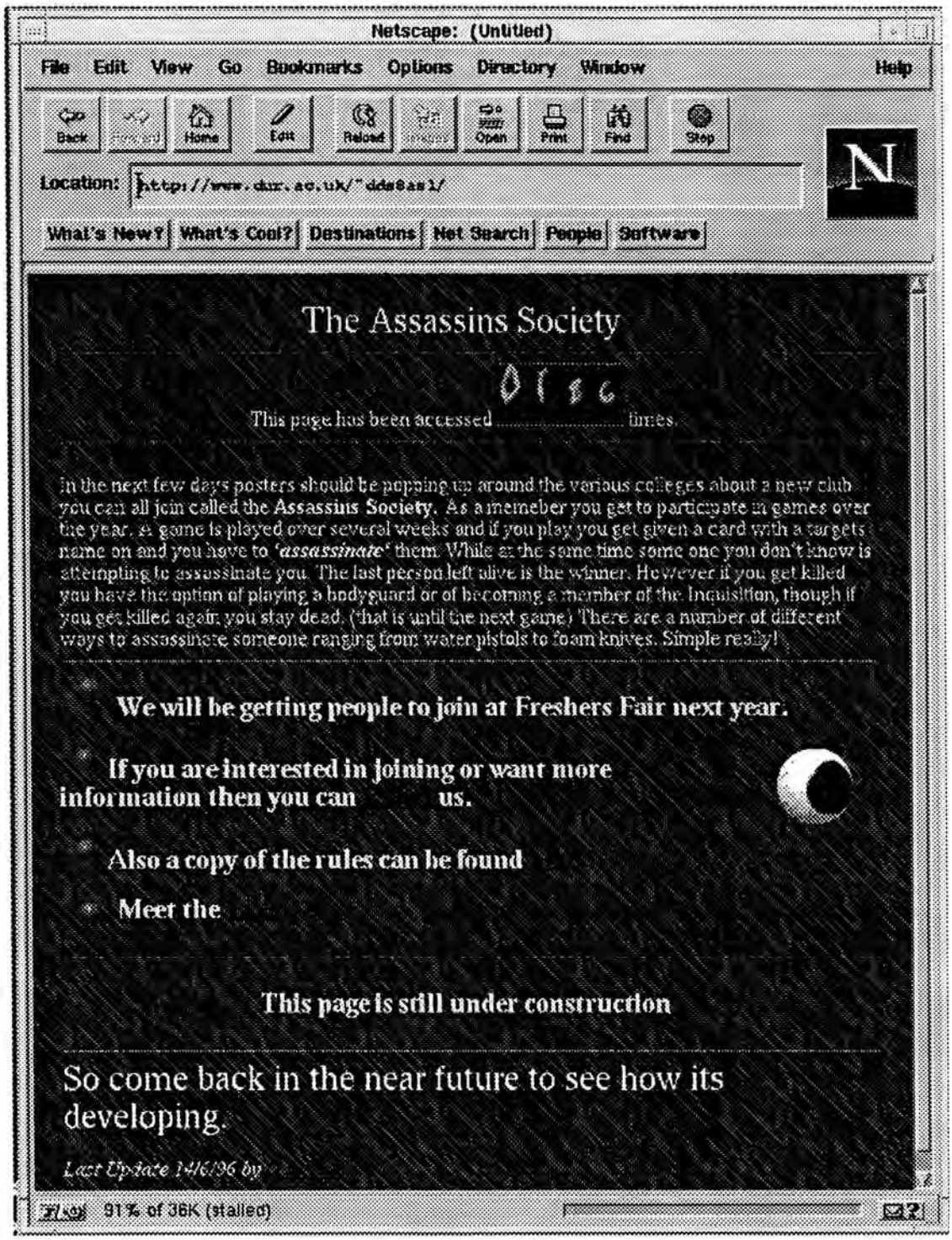


Figure 4 An example of a possible background

All these activities need to be addressed in order to have good quality pages. This is why a good approach to maintenance is required. Since many hypermedia applications have volatile data that requires frequent updating, there is a need to standardise and automate the development and subsequent maintenance [16].

2.4.3 Current Problems

The original aim of the World-Wide Web was to aid distributed working through information sharing and collaborative authoring using a simple, uniform medium. The very nature of the Web, however, has led to some of its problems. Hyperdocuments on the World-Wide Web are often written by several authors and maintained by a number of maintainers. This means that the hypertext application is often distributed across a number of sites making the maintenance activity more complex relying on the diligence of each document owner [9].

As with all aspects of the World-Wide Web there is a lack of tools for maintaining Web pages. There are many tools for converting text to HTML, as well as a number of HTML editors, unfortunately there are few that help develop good quality pages and none really address maintaining the content of the page or the structure even though some will check the links or the validity of the HTML [12, 16, 3]

There are also problems which occur due to the nature of hypertext systems. Hypertext systems tend to have difficulty with rapidly changing information due to the essentially static nature of the hypertext model in which the network does not change unless it is explicitly edited by the user or some other external agent. In particular, the network cannot reconfigure itself in response to changes in the information it contains. The lack of dynamic mechanism limits the utility of hypertext in many tasks. [18, 17]. Off-line methods are unsuitable as the links need to be continuously and rapidly updated [17].

The areas that need to be addressed by maintainers are structure and quality problems, maintenance documentation and integrity problems caused by making changes.

2.4.3.1 Structure Problems

Many authors have addressed the problems of structure and its importance to hypertext. Poorly structured documents lead to difficulty navigating the hypertext application giving a feeling of disorientation. The hypertext structure itself expands exponentially with the continuous linking of nodes, which creates more orientation and navigation problems. Orientation and navigation have to be addressed with the increasing size of the Web [12, 2, 3]. A good structure is important as it improves navigation [2], as well as maintainability and readability [12]. The underlying problem being that good structures do not just exist, they exist by design [9]. Unfortunately authors of Web documents often do not design good structures.

Authors, at times, do not understand the concept of a hypertext application. They essentially structure the document often like a copy of a paper document, not making use of the technology. If links are inserted they often do not follow any structure but are merely haphazard [12]. As individuals make changes to improve their own structures, they may then break others' structures. The problem is also exacerbated by scale, as the structure grows, it becomes more complex and difficult to maintain.

Hyperdocuments are not just individual documents owned by a single author, they can comprise of varying information sources and interlinked documents that are at distributed sites. These are often maintained by a number of different document authors also often at distributed sites (and the maintainers are not necessarily the original authors). Such a structure is known as an infostructure. These infostructures are rarely static and their contents may change, thus changing the intended structure. Documents are often moved, deleted, referenced information may change and hypertext links then become broken [9]. Maintenance currently relies on the diligence of individual authors, complaints of users and error logs of servers. It is unreasonable to expect the user to maintain these links manually because of the cost in time would be prohibitive and the user cannot be guaranteed to do the maintenance correctly. The result of this is that maintenance, when performed, is often not correctly and completely done and the structures will become corrupted.

2.4.3.2 Quality Problems

Documents on the Web are all too often of poor quality. This could be due to two problems. There is no incentive to produce good quality documents as these documents do not go through the same channels as normal publishing procedures of reviewing and formal editing applied to books and articles. The documents are not subjected to criticism and rejection, making it easy to overlook the issue of quality. The second problem is not as easy to solve. There appears to be little consensus on what good quality is [16, 13], let alone standards and guidelines to help guide the maintainer in assessing what should be altered or not.

2.4.3.3 Documentation Problems

The primary source of maintenance information is from users. This is commonly in the form of complaints from those who encountered broken links or malformed documents. However, information providers cannot rely on complaints. Most users of the Web are very tolerant of the errors encountered, after all, since the information is often provided by volunteers, many

users feel that it would be impolite or disrespectful to point out any problems. Furthermore, many documents do not explicitly indicate to whom complaints should be addressed and even when a complaint is received it is often directed to the wrong person (i.e. the one maintaining the source of the link instead of its destination).

A second source of maintenance information is provided by the log files of each server. The server records, or attempts to record, each document request and, if an error occurred, the nature of that error. Such information can be extremely useful for identifying requests for documents that have moved and those that have misspelt URLs. However, only the server managers have access to this information and the error is often never relayed to the document maintainer, either because it is not recognised as a document error or because the origin of the error is not apparent from the error message. Logs also cannot reveal failed requests that never make it to the server nor can they support problems of changed document content [9].

A third source of maintenance information is from analysis tools. These tools can provide useful information on broken links or invalid HTML. Unfortunately, most of these tools are very specific, only performing a minimum number of functions and usually being tailored to a specific user's needs.

2.4.3.4 Integrity Problems

Servers and sites often start life as an experiment and their authors often do not give much thought to the issues of naming documents and servers, later it often becomes necessary to reorganise their resources. So they rename their resources, which causes problems in broken links with documents within the system and other documents linking into the server. One should treat WWW as a massive file system.

Unfortunately there are a number of restrictions on preserving the integrity of a site. Even if there were guidelines and standards for the Web, there is no mechanism for enforcing restrictions on the Web relying on volunteers to enforce them. This problem can only be solved by guidelines and standards set up and enforced by the server administration.

In order to avoid the problems of integrity, an author would have to always give thought to the name of the document and not delete that object once placed on the server in order not to create broken links for those linking to the object. The only possibility for removing the object would be to contact all those who have linked to the page which is infeasible. Unfortunately, there is

traffic would still be increased. There would no longer be the problem of relinquishing control to a single node and thus less likelihood of the system being rejected by the security conscious [8].

Problems with this arise with the ordering of events and mutual exclusion of concurrent systems. Lessons from distributed operating systems, distributed DBMS and managed networks would need to be learned.

2.5.2 Analysis of Hypertext and World-Wide Web Documents

It is perhaps not always appropriate to use software techniques when working with hypertext applications. In the case of [19], they advise not to use software evaluation techniques for evaluating hypertext applications. They do not consider it good to evaluate the system by how well it conforms to user needs either, a main concern of usability evaluation techniques.

2.5.2.1 An Approach to Hypertext Analysis

The approach defined here is done by deciding on the high level criteria which are needed to determine good quality, these have been gathered from [19, 13, 12]. These high level criteria, often being difficult to measure, are broken down into lower level criteria to make it more easy to evaluate. From there, if a little crudely, an application can be evaluated.

2.5.2.2 High Level Criteria

In [19], the criteria required in order to make an application of good quality were: richness, ease, consistency, self-evidence, predictability, readability and reuse. Richness expresses the abundance of information items and ways to reach them. Ease measures information accessibility and how easy to grasp operations are. Consistency measures application regularity and can be summed up by a generic rule, which is treating conceptually similar items in a similar fashion and conceptually different ones in a different fashion. Self-evidence expresses how well the meaning or the purpose is of what is being presented, can be guessed by the user. Predictability expresses how well the user can anticipate an operation's outcome. Readability expresses the overall "feeling" about an application's validity. Readability also depends on all factors mentioned. Reuse considers using objects and operations in different contexts and for different purposes. Reuse promotes consistency (and therefore predictability). Also, presenting the same material under different perspectives and points of view enhances the applications richness. Additional reasons for reuse can be found among standard software engineering

argumentation: it reduces mistakes, development effort and physical space required. Reuse can expand an application's apparent size, while minimally increasing its physical size.

[13] takes a different approach to defining his criteria, stating that the greatest need for hypertext in 1990 was for authors to exploit the medium properly and successfully. In his paper he uses the following criteria: the hypertext application is of immediate appeal to readers, it must have a potentially long lifetime, and the hypertext application can be maintained and enhanced by other authors.

In 1990 [13] stated that poor quality of authorship arises due to inexperienced authors and crude tools. [13] felt that the assessment of hyperdocuments needed to be done by a referee providing feedback The referee needed to be a type of editor, a teacher or an expert in hypertext and in presenting material using hypertext.

In [12], they attempt to measure the readability and maintainability of hypertext. To help determine the quality of a hypertext application, a measure of it is first required, which is what is attempted in this paper. To determine quality of hypertext applications, it is necessary first to identify high-level factors which determine quality: readability, maintainability, correctness, integrity, usability and testability. Readability factors address whether it is easy for the reader to get lost in hyperspace, whether there are maps and facilities available and whether the content of the nodes are correct and easy to understand. Maintainability asks whether there is design documentation available, whether there are any comments of the author on the structure of the document included in the hypertext application and whether it is easy to understand the intentions and the authoring style of the original author.

So overall, the high level criteria were as follows: readability, maintainability, testability, reuse, consistency, correctness and usability.

2.5.2.3 Criteria for Evaluation

It is not possible to assess outright what is a good level of maintainability or the other high-level criteria. This is because there is no specific method for measuring such criteria, as there is no unit of measure. Thus all three papers [19, 12, 13] chose to take lower level criteria which were measurable in order to ascertain whether the application in question satisfied the above criteria.

[19] chose the following low level criteria: content, structure, presentation, dynamics and interaction. Content being the pieces of information included in the application. These consist of static media and dynamic media (video clips, soundtracks and animation are some examples). Content analysis may be the most important dimension for assessing a hypertext application's effectiveness. This would require an application domain specialist and a deep knowledge of the users' profiles and tasks. Structure describes the content's organisation. Presentation is how an application's content and functions are shown to readers. Presentation includes visualisation of individual granules of information and dynamic features such as navigation. Dynamics are how users interact with individual pieces of information and move among them. Interaction is using an application's dynamic functionality through operating on presentation elements. Interaction blends both dynamics and presentation.

[13] suggested that the evaluator answer the following questions about the application.

1. How big is it? How much is text and how much is graphics? How many files are used?
2. Does it contain spelling mistakes or grammar problems?
3. Is its structure logically correct, e.g. is it connected? How many hierarchical links, how many cross-reference links are there?
4. Which facilities does it use?
5. What is the average number of links to an atomic segment of hypertext? This is called regularity, by analogy with VLSI design. (The regularity would be one for a sequential text or a purely hierarchical one; for a text full of cross-references, the regularity would be higher.)

This gives [13] the preliminary assessment of hypertext. They give general guidelines for the referee. The above can be mechanically done by the computer.

Other helpful points were:

1. Is the work generally static, not consisting of tables or pictures generated on-the-fly?
2. Is the work self-contained, not making use of shared global material? It is common place on the WWW for use to be made of shared global material.

[13] found that using the above criteria, a general idea of the documents quality could be assessed. For example, a larger document will have more problems, a document with spelling mistakes is likely to be poor in other areas or if the document is badly structured, it is less likely to be readable.

Criteria for assessment:

1. How well does the hypertext application do its job? How effective is the presentation?
2. Quality of design
3. Use of facilities
4. Quality of text/graphics
5. Structure
6. Navigation

[13] warns to be careful not to focus too much on the details and forget the big picture.

[12] defined hypertext application attributes as being size, path complexity (linearity and cycles), tree impurity, modularity, individual node complexity, coherence, complexity of node contents and simplicity. It should be remembered that [12] geared their low level factors towards metrics for measuring maintainability and readability.

Broadly speaking, through reading the three papers, low level factors can be grouped to form a set of factors which span the three papers. [12, 13] consider size to be a factor. [19, 13] consider structure as a low level factor, however, [12] takes it further and divides up the structure into path complexity, tree impurity, coherence and modularity. [19] states that content is a factor, while [13] talks about effectiveness, use of facilities and achievement of goal, which though they involve other factors too, all relate to content. [12] use complexity of node content only. Presentation is covered by [13] as quality of text and graphics and [19] also regards presentation as a factor, however, [12] do not address it directly. The final area is navigation. [13] considers navigation a factor, while [19] talks about dynamics and interaction. So overall the main low level factors to considered were: size, structure, presentation, content and navigation. These factors can of course then be divided further into lower level factors.

2.5.2.4 Common Faults

When carrying out reviews of hypertext applications, [13] found that most hypertext applications were substandard and that many faults appeared regularly. He found that it is good practice to watch for these faults. Many of the following faults still appear regularly on Web applications.

Over use of the technology is a common problem, it often occurs that a lot of use of special effects is made with very little substance in the actual application. The visual appearance is frequently, poorly designed. This is often the case when the designer is inexperienced with

hypertext, which can also lead to the designer not exploiting the medium properly and producing an application which is merely a copy of the original paper document. Anchors in a hypertext application should give an idea of what the destination document will contain, unfortunately use of words such as 'this' to refer to the anchor is a regular occurrence. Lack of a good design leads to a lack of coherent overall structure and presentation style. Many documents are poorly written with bad grammar and many spelling mistakes. Many hypertext documents are written in such a way that the author does not take note of the fact that the document may be viewed with a different browser or with a different size window, altering the whole effect of the document. Material is often badly structured, with the same information appearing twice, with material being difficult to follow in a logical order and sometimes with documents being linked together in many different places.

2.5.2.5 Use of Metrics

[12] pursued the idea of developing metrics in order to assess problems with hypertext and to ascertain the level of quality, as well as readability and maintainability. The reason behind using these metrics is so that the maintainers can predict and plan for the next phases, particularly testing and maintenance, identify areas of hypertext that are too complex and serve as a component for a quality model.

The focus for the structure metric is the hypertext graph. Hypertext metrics have already appeared in literature [13]. The metrics are derived from well known software metrics. A hypertext graph uses directed graphs. Path complexity is the number of different paths or cycles which can be made. For structural analysis of hypertext graphs: compactness and stratum metrics are suggested. The path complexity criterion is the compactness metric, which is how easily a node can be reached from another node. The stratum metric is the linear ordering of the document. A well known software metric for path complexity is that of McCabe's cyclomatic number:

$$V(G) = 3D\ e\ (\text{edges}) - n\ (\text{vertices}) + p\ (\text{connected components}).$$

Measuring size is important as the bigger the hypertext application, the increased difficulty in maintaining and reading. For program size the LOC (lines of code) unit is used as an intensive measure. Here the number of nodes and links are proposed as a measure, however, this doesn't take into account the size of the node.

2.5.2.8 Problems

[13] made the following conclusions about the problems involved in analysing hypertext applications. Quality assessment can only be done by humans, knowledgeable humans with a good understanding of the application domain. There is a need better tools to test coverage analogous to testing programs. There are numerous pitfalls and so referees are needed to prevent authors from falling into these traps.

2.5.3 Software Tools and the World-Wide Web

There are to date, numerous software tools for use with the Web. Indeed since the start of this research, many have appeared on the market or on the Internet. The time and resources required to be able to develop a comprehensive study of them all, that is also up to date are beyond the scope of this thesis. The tools that are available perform a variety of tasks, HTML editors, HTML generators, Web authoring, HTML checkers, link checkers, generators from HTML, HTML analysers, server log file analysers, web wandering robots, bookmark managers, graphical tools, imagemap tools and site management tools. As the main focus of this work is Web maintenance, tools that are not relevant to Web maintenance will not be covered in this section. It is also important to note that not all maintenance tools will be mentioned but more of a sample of maintenance tools will be given in order to give an idea of what the currently available tools do and what is still needed. One final note before commencing, there is no specific focus on any platform. The main tools that are important to Web maintenance are HTML validators, link validators, server log file analysers and site management tools.

2.5.3.1 HTML Validators

HTML validators are tools which check the syntax of the document and return a list of errors. There are many forms of HTML and many tags which are browser dependent, hence HTML developers need to check their syntax in order to assure themselves that their documents will display correctly on all browsers.

A Kinder Gentler HTML Validator

This validator allows the user to fill out a form and send away for the results. It uses a program called "weblint" which is a perl script for checking HTML (HTML 2.0). This gives very easy to understand explanations of the problems which have occurred and is very helpful for those who don't know much HTML. It does not however, appear to be as sophisticated as many other validators and appears only to check the syntax.

WebTechs HTML Validation Service

WebTechs validation service is a form based service, which has the advantage of being easy to use and the user does not have to download and install software. Unfortunately this has the disadvantage of being slow or unreachable at times due to network problems. WebTechs performs syntax analysis, produces a list of tags used and can be channelled to a specific browser such a Netscape 2.0.

2.5.3.2 Link Validators

Link validators test to see that all the links within a document lead to another document or resource. One of the big problems with maintenance is that of broken links and manual link checking has proven to be difficult, tedious and time-consuming. Link checkers fall into two categories, those that only work locally and those that check links globally.

html_analyzer

The html_analyzer is useful as it not only validates links, it also analyses the HTML. This is a piece of software that has to be retrieved and installed, however, it is public domain and can be retrieved by anonymous ftp. The intent of the tool is to assist the maintenance of HTML databases. The authors recognise the need for automated link validation as the size of the database increases. However, this tool is not capable of validating non-local links. The software extracts all the hyperlinks (i.e. anchors) from all the *.html files within the directory hierarchy. The software validates the availability of referenced links, with a test call "validate". It then checks the hyperlink contents that occur in the database, but are not themselves hyperlinks and runs a test called "completeness". This is followed by a test for consistency, checking one-to-one relations between hyperlinks and the contents of the hyperlink.

MOMSpider

MOMSpider solves the problems of manual traversal of the Web for testing and maintaining Web pages. Current Web browsers are designed for normal viewing only and make no distinction between old documents and those that have recently changed, nor do they show the user last modification or expiry dates. The only method for testing a link through a browser is by transferring the whole document. This is very inefficient, which can deter people from testing links. MOMSpider can test the links without transferring the whole document, it does this by using HEAD request method rather than GET used by browsers. If document meta-information is available in the headers other special conditions can be tested as well, such as last modified and expiry dates. The program can furthermore restrict its focus to the Web

structure rather than the contents of each document. An automated traversal program should be able to test multiple infostructures maintained by different people and share the information across them. This reduces the overhead of maintenance and Web traffic as the tool can provide maintenance information for lots of people. The best place to put results for multiple owners is on the Web itself. [9] argues that no automated traversal program can completely solve the problems of Web maintenance, for example, a program cannot tell when a document's contents have changed and they no longer represent the infostructures intentions nor can a program, once it has discovered a broken link, determine why it was broken or how to fix it. These tasks will still be performed by hand by the maintainers. MOMSpider is a Web wandering robot, that, given a list of instructions, that details what infostructures to traverse, whom to notify for problems and where to put resulting maintenance information will traverse an infostructure and fulfil all of the requirements listed above. MOMSpider was specifically designed to fulfil the problems of multi-owner maintenance. Other requirements were to minimise its effect on WWW servers and network bandwidth. It respects current limits placed on Web wandering robots. The four types of information that MOMSpider would deliver to the owner:

1. referenced objects which have redirected URLs (moved documents);
2. referenced objects which cannot be accessed (broken links);
3. referenced objects with recent modification dates; and
4. owned objects with expiration dates near to the current date.

EIT Link Verifier Robot

The EIT link verifier is part of a suite of tools, this tool is intended for maintaining links within the documents managed at the maintainer's site. This tool has the advantage of being able to test non-local links. The link verifier tool starts from a given URL and traverses links outward, subject to a specified search profile, producing a report of the state of all discovered links. The EIT link verifier has the following features: a tailored search profile which allows the user to override default form values or to restrict the output report. It has a parallel search which can be employed to verify remote links. It does background searching if an e-mail reply is specified. Incremental searches can be achieved by specifying a time-out. Form POST actions will be verified up to a point. All buttons in an HTML form will be pushed.

URL Minder

This link checker has the advantage that the user does not have to physically go and check the page's links. This one will check the links for the user and e-mail any problems back. It also has the advantage that the user does not have to acquire any software. This software works by

subscribing to it via e-mail with the address of the URL that the user wishes to check the links of. The URL-minder then retrieves World-Wide-Web resources from time to time and checks to see if they have changed from last time they were retrieved. It sends the user e-mail when the Web pages the user has registered change. It also performs a Yahoo search, if the user copies the URL of the search result and paste it to their URL-minder registration form, it will work for any search engine that uses GET. The URL-minder is useful for keeping track of the user's pages and any changes which occur. It also helps keep track of competition for the user by running keyword searches on a regular basis, it checks to see if anything new turns up. It can also be useful for finding out when it is difficult to retrieve the user's pages from the outside world, a fault that can often go unnoticed. Only URLs with "http" are valid for the URL-minder. The URL-minder also contains a time-out facility so very large files on very slow servers may not work. This, however, can be useful in letting the owner know that his pages are difficult to access. A problem that could arise with the URL-minder is the regularity of checks. Currently it checks pages in the region of once a day to once a week, however, this may be much longer as the number of subscribers increases.

2.5.3.3 Server Log File Analysers

These tools are only really relevant to server administrators, however, they contain information that maintainers may often need to know but are not privy to. In order to ensure the smooth running of a site it is important to pass on this sort of information to the persons concerned.

Getstats program

This sorts of program is very useful for maintaining sites. It takes the log file from a CERN, NCSA, Plexus, GNNPress, MacHTTP or UNIX Gopher server and gives back a variety of statistics. Getstats is not public domain. Getstats produces twelve different reports and the user can request any combination of these reports. These reports are, daily, hourly monthly and weekly, either full, domain, request, directory tree, file or error reports. The program can produce statistics such as number of requests, number of new unique hosts, total number unique hosts, number of HTML requests, non-HTML requests, script requests, malformed requests, total number of requests, the average number of requests per minute/hour/day and the running time of the program.

2.5.3.4 Site Management Tools

As the Web grows and grows, it becomes larger and more complex, tools for automatic site creation and management may well be the next area for a development explosion, however,

there is comparatively little in this area at the moment. One trend is for editor-server combinations such as the proprietary GNNPress/GNNServer and Microsoft FrontPage, and the non-proprietary Amaya and SiteMill. The second approach is that of Clay Basket for the Macintosh, which combines with the scripting language (Frontier) to provide a template-driven facility for automatic HTML markup and site update.

2.5.3.5 Advantages and Disadvantages of Current Tools

Web maintenance tools generally come in three formats: WWW form-based services, e-mail based services or software tools which the user can buy or download and install on their own machine. Each type of tool or service has its advantages and disadvantages. Form-based services are simple to use, the user does not have to make sure that the software is maintained, users do not have to acquire the software or use any of their own resources to run the program. The disadvantages of these are, if the network is busy or down the service may be unreachable, these sorts of services can also pose security risks with the transmission of data over a network. E-mail based services have similar advantages and disadvantages, however, they do have the added advantage of often being services that are run remotely without the user having to be there while they work. Software tools do not provide the problems that occur with networks, but they are also limiting in that they cannot for example, check non-local links without the use of the network. Software tools can be expensive and use a lot of the system resources too. Each type of tool has pros and cons, however, none solve the problems comprehensively.

2.5.4 Visualisation and Navigation of the World-Wide Web

Navigation is an important part of Web maintenance. [12] states that navigation is not only important for readability, if there are difficulties in navigation, this leads to difficulties in maintenance, if the user makes a change to a hypertext application it is important they understand the structure. Unfortunately, navigation problems are common to hypertext and user disorientation has been recognised as a big problem [2, 20]. Navigation is important for other reasons too, when browsing, sometimes the journey matters more than the destination. The user can get useful information merely by travelling through the hyperspace. Users also need to know at any stage, where they are and how they got there.

The main solution provided so far to improve navigation has been through better structuring of the document. According to [2], tree-oriented hierarchies give better navigation.

[16] recommend that high level of access be structured by grouping items of interest. This design can be bottom-up or top-down. They also recommend that the depth of hierarchies be reduced to limit user disorientation.

[21] introduces the concept of local and global coherence and navigation. Global is moving within a collection of information that spans many nodes and local is moving between pairs of specific nodes. For good navigation the designer of a publication in hypertext must support both concepts. HTML doesn't distinguish between local and global navigation. It provides three pieces of orientation information: (a) the title of the current document; (b) the Universal Resource Locator (URL) of the document; and (c) the URL of a link destination. These do not support the users' mental model. A common strategy to overcome this is to establish a graphic set of symbols that will help guide the user's navigation and tie together the collection as a whole. Local coherence is addressed by asking such questions as: Where are all the links and anchors in the document? When do I follow this link? Where am I going? Why am I going there? Local coherence is controlled at semantic level in HTML documents, links being anchored at words, phrases and graphics in browsers, by the use of colours or underlining, this gives a visual distinction between links and non-links. The concept of a link explainer is not supported. The lack of support for the local coherence makes the need for global coherence all the more acute. The designer must help the user answer questions such as, what collection am I in? What part of this collection does this document represent? It is apparent that the designer of hypermedia applications must struggle with the mechanisms of the system itself to produce local and global coherence.

The second solution, which helps provide support for local and global coherence, is quite prevalent, being that of providing maps of the hierarchy, clickable or otherwise. The solution is extended further by [20] which includes ideas on how to visualise these maps. Representations of documents in a content space is done using a ball to represent a page or a collection of pages, while straight lines represent hyperlinks. To represent the documents in a content space, documents of similar content would need to reside near each other. A tool has been developed at Birmingham University, called Hyperspace [22], to help with navigation. Unfortunately it is still in its early stages and so is limited in its usefulness. It allows the user to view a map of where they have been and where they are when they are browsing the Web. Unfortunately, it is difficult to distinguish specific elements in the graphical representation. [16] also advise using a clickable map as a means of navigation to help the users see where they are and navigate their way round, with pointers in the form of buttons.

A third idea proposed by [20] is to allow the users to create their own structures on the content map, by being able to point at a document and drag it to whichever part of the screen they choose.

In summary, the work done so far to improve navigation is based around better structuring and the visualisation of hyperspace. Unfortunately, although many of the solutions are workable, they are not totally satisfactory as problems still exist with user disorientation and maintenance of bad structures.

2.6 Summary of the Main Points

The World-Wide Web came about to solve some of the problems that were being encountered by scientists trying to work collaboratively. The problems it solved were those difficulties previously encountered through accessing and sharing information over a range of different protocols and platforms. The result was a user-friendly Internet information service.

Currently the Web is developing quickly and is used by all kinds of organisations and institutions for a range of different purposes. The advantages of the World-Wide Web are that it is easy to use and due to its popularity, it is now an important information source. The Web has a number of problems too. There is a lack of adequate tools for use with the Web. The problems of hypertext, such as disorientation have not been resolved and thus have been passed on to the World-Wide Web. The Web is still complex to use. A fast evolution has meant that the Web has not developed properly and constraints still exist. There are few and for the most part no standards and guidelines to help users to improve quality. Due to the lack of standards and tools, problems of authorship quality have arisen.

In order to improve the problems standards and guidelines are required and an improved approach to maintaining Web sites may improve quality. Web maintenance is the process of updating a page regularly in order to keep it relevant and maintain its quality. Web maintenance is needed as most applications designed for the Web contain volatile data that requires frequent updating. Structural problems with pages can appear where links have become broken and in some cases the quality of a page may need improving. The problem areas with Web maintenance are lack of proper structuring, poor quality, lack of maintenance documentation and integrity problems caused by making changes.

The previous work done in the area of Web maintenance is in enforcing standards and guidelines, analysing hypertext applications and the creation of tools for the Web. These tools include HTML validators, link testers and log file analysers. Work has also been done to improve navigation which has been based on better structuring of the Web document and visualising hyperspace.

In the following chapter, two case studies are discussed. These case studies and the problems identified in the current chapter form a basis for a discussion and conclusion which lead on to the development of a proposed solution.

CHAPTER 3

CASE STUDIES AND PROBLEM DEVELOPMENT

3.1 Objectives of the Chapter

The chapter takes two case studies and examines the good points and the problems encountered with managing and maintaining a Web site. It then goes on in the next section to discuss the problems identified in chapter two and the problems found through the case studies. This synopsis of the problems is followed by a discussion of the ways in which this work could contribute towards solving some of the problems.

3.2 Case Studies

In order to gauge what current maintenance practices are, two case studies were carried out, one with the CCTA and the other with Durham University IT Service. Both case studies were carried out by talking to the management team and looking at the Web sites of each group. It was felt that carrying out case studies would provide a better idea of whether maintenance of Web sites is regarded as an issue and if so are the organisations in question encountering problems.

3.3 Discussion on the CCTA Web Site and Management Procedures

This section gives a discussion of the management and maintenance procedures used at the CCTA. It starts by discussing the good points, which divides into maintainability, readability and quality attributes. The next part goes on to discuss the problems encountered with the procedures. These are broadly split into two categories, recognised problems and other problems. The first section is divided further by looking at maintainability attributes and

readability and quality attributes. The problems sections look at technical and managerial problems separately.

The CCTA has a branch known as the CCTA Government Information Service (CGIS). This part of the CCTA hosts Web pages for public sector organisations. The CGIS team consists of six members, all working on various aspects of the operation and maintenance of the Web service. This ranges from maintaining and operating the server to researching emerging technologies for enhancing the Web.

The CGIS team have only one point of access to the site. This means that they have a centralised control of the site. The CCTA teaches members of the organisations how to develop Web pages and then accepts from them the pages and places them on the Web. At the CCTA, they set out a number of in house guidelines and standards to the organisations. For a full description of the CCTA procedures see appendix B.

3.3.1 Good Points

The following section represents beneficial aspects of the current system with respect to its maintenance. The section has been divided into readability, quality and maintainability attributes. These attributes should be maintained and built upon.

Maintainability Attributes

The CCTA have a central server which keeps a clear focus on the big picture. The pages are well structured. They have a booking-in system for the pages to ensure that pages that have arrived, do not get lost or neglected. This will also give a good record of the pages that are on the Web and other statistics such as the growth rate of the site.

The CCTA site does not have the problem which occurs at other larger sites where many people have access to the UNIX server and can put pages on and off the Web as they please. This present problems with preserving a well structured or maintained site.

It is debatable whether having only one means of access to the Web is a good thing, however, it does keep the site in a good readable and maintainable state, if a little static. They have given thought to maintenance in that they have developed a tool to solve certain problems, see appendix B.

Readability and Quality Attributes

The CCTA's pages have a certain relevance. The organisations are encouraged to target their audiences. They have created some standards and set out guidelines that will decide whether a document is good or not. This helps in the development and maintenance of Web pages and does not usually exist in the development of other sites. There is help with the classic problem of disorientation through the use of a good structure, an index of the organisations and a button to return to the home page of the CCTA. The contents of the main document are concise and easy to understand. The system is 'user-friendly'. It would be quite easy to check the paths through the site as they are tree structured as far as possible and the pages are coherent, they have the same 'feel' mostly keeping a certain uniformity.

3.3.2 Problems

The following are some points, which I felt were problems that the CCTA Web site had and in certain cases were not aware of. I have split this section into two parts one listing problems that are recognised and the other problems of which they are unaware at present.

Recognised Problems

Technical Problems: DOS and UNIX file naming conventions

There is a problem with checking filenames on DOS, in that it is not case sensitive, unlike UNIX, and thus, mistakes of case will not be detected causing broken links. The problem with checking files in DOS is that it does not pick up errors in filenames. For example the '.HTML' extension is too long so it will only accepts '.htm' and the case of letters in filenames are not picked up in DOS but are on UNIX!

Managerial Problems

Some consistency problems were detected. There is a problem with providing different services depending on the platform available. If a change is made to one service, then all other services should be altered too. This unfortunately does not always happen. In the case where the buttons that are standard to all pages, what would happen if the buttons changed? All the current pages will have to be changed and in the case of independent organisations how will these be changed? Finally, the standards are not enforced, allowing people to deviate from the 'CCTA feel'.

Other Problems

Technical Problems

There are some technical problems with the guidelines provided to the developers. Centring pages is difficult as this is a Netscape extension and some browsers don't support Netscape extensions. Requesting that a front page fit onto the screen is difficult to monitor as there are some very small screens around. Unless the person concerned is an expert in this particular field it is difficult to assess what is good and what is not on all screens.

The CCTA should also request context-free document titles. Documents do not always have this. If a person arrives at a page without going through the home page it should be clear from the title what the document is about. This could cause confusion to the reader.

Managerial Problems

They don't request permission from organisations to make the changes. This could potentially cause a problem if the organisation is not pleased with the change. The CCTA carries out no quality controls on the contents of the documents, bar their own guidelines. They view whatever is contained in the document as the problem of the organisation. But it should become something of a problem to the site manager too. Once the page is on the Web, no maintenance checks are carried out. The length of time to put a new page on the Web, is estimated at two days, this would seem long, considering the nature of Web, changes should be quick. The Web site manager should be responsible overall for the site and should therefore check for major problems. Most of the authors involved in the development will have very limited authoring experience. Authoring is often perceived as merely the ability to write HTML, but many other factors are involved such as, for example, structuring of hypertext. Insufficient use of tools is made. A tool for writing HTML and a link testing tool still limits the automated maintenance, thus increasing the overhead of work involved. Analysis of log files is done, however, major errors are not reported to the owners of the documents.

3.3.3 Points for Consideration

These are some points that are important and should be considered when developing and maintaining a Web site. In as far as I could see the site is well distributed. Search engines pick up the CCTA home page quite easily when using the keyword CCTA. However none of the following points were mentioned. A Web page developer should always bear in mind the reason why the Web page is being created and future developments should not change the purpose of the page. Information presented should keep its relevance over time. The site should not contain any unnecessary repetition of information. It is important to keep a site of reasonable size. As a site increases in size, the maintainability is reduced. The CCTA site is currently a manageable

size which makes it quite maintainable. The growth of the site could make it unwieldy and thus reduce the maintainability. Removing pages from a site can cause numerous problems with broken links. Pages at a site often link to each other. The CCTA have not currently had to deal with the problem of pages being removed.

3.3.4 Suggested Improvements

The following is of list of improvements that could be made to the CCTA site in order to make the maintenance of the site more efficient. Some of the suggestions are straight forward to carry out, others require more work into the field of maintenance.

Straight Forward Improvements

Maintenance checks would be advisable periodically for the links and graphics at least. It would also be advisable to look out for pages that have not been updated for a while or pages where the information has become irrelevant. The maintainers of the pages should be alerted to this fact. It is the responsibility of the Web Site managers to ensure that the whole site is properly maintained even if they pass on the maintenance activities needing to be carried out to the organisation concerned. A periodic maintenance check should be carried out and should include checking that the page has been regularly updated. For navigational purposes it would be good to have a button that will take the user back to the home page of the organisation. Checking documents on the mirror site is not enough. The documents should be checked on the UNIX platform too to eradicate errors that have been missed in DOS. Changes made to standards or structure of the CCTA pages should be communicated to organisations running their own services. An example is the change in appearance of the help button or the addition of a feed back button. The organisation can then choose to add them in too to their own pages keeping the 'CCTA feel' to their pages. The CCTA should use of the information obtained through analysing log files of the server and then communicate the problems found to the developers of the page.

For Future Consideration

No mention of teaching a development method to organisations was made. It is often a useful idea to encourage a development method with a design. This will help eradicate problems such as badly structured documents and thus improve the efficiency of maintenance. Any changes made to pages should be documented in order to help the maintenance of the pages and to help in tracking problems. Some configuration management would help in making sure that all services for varying platforms are up to date and there are no inconsistencies. A process which

could be automated, for requesting permission to make changes should be in place. The process of putting new pages on the Web should be quicker. Automated checking of pages could produce a quicker response time by reducing the work overhead. Some monitoring of the quality of the content of the documents should be carried out in order to ensure that what is being put on is, for example, relevant, correct and useful. Developing a design method or template for the organisations to use when developing their own pages would be a help to inexperienced developers. People are less likely to try out their ideas if a template is present and thus this could also reduce the maintenance problems. The CCTA could investigate further the use of readily available tools for maintaining the Web.

The Issue of Non-Professional Developers

One of the major problems that the CCTA is faced with when getting this service off the ground is that the people who will ultimately be developing the pages are not software professionals. They will therefore not take into consideration the concepts of development and maintenance and consequently produce more difficult to read and maintain pages. These problems be compounded with the increased use of Java and VRML.

3.3.5 Foreseeable Problems

As the site grows and develops other problems will occur. Below is a list of some of the problems that are predicted to occur in the future. These problems will not all necessarily be eradicated through better maintenance, however, some will and these are addressed first.

Problems that can be solved through better maintenance procedures

Configuration management will be needed to help keep track of versions, particularly, when they offer alternative services depending on the platform. The standards are not rigid or rigidly applied, they are merely advised. Applying the standards more rigidly could reduce the maintenance overhead. In the future, as the site grows, the uniformity and general feel of the pages will be lost. If developers are allowed to deviate from standards and guidelines, it will become more difficult to keep a track of the pages. When the use of Java applets is put into practice, these standards will have to be very rigid, otherwise, there will be the classic problems that occur in software engineering. A Web site of bug-ridden applications does not present a good image.

Other Problems

Eventually maintenance costs will outstrip development costs. This is a common feature of maintenance. The structure of the site will expand exponentially, becoming increasingly difficult to maintain. The infostructure is also becoming increasingly distributed and thus the number of authors increases, the complexity increases and the maintenance becomes tedious. Soon, maps of the structure will be necessary to help users navigate the system.

Currently the staff are learning to program in Java and they will initially write the applets for the customers. Will this bring in the classic problems of software maintenance and software engineering? Are they equipped to do things like requirements gathering of the organisations? Java will probably also imply a serious revision of their standards.

3.3.7 A Summary of the CCTA Government Information Service

This service is well structured giving much thought to development and maintenance of a Web site. There are still some problems such as not having regular maintenance check of current pages, however, this problem and other problems described can be eradicated quite easily. A more general problem is the lack of standards for the World-Wide Web. This is not just specific to the CCTA and must be addressed more widely through research.

3.4 IT Service Case Study

This section describes the case study carried out with the IT Service. This was done through talking to the manager of the Web site for the IT Service and through looking at the University pages.

3.5 Discussion on the ITS Web Site and Management Procedures

A full description of the ITS Web site management is given in appendix C. The following is a summary. The IT Service has two sorts of Web page. The first is an official page which is linked to the university pages and is part of the site which is actively managed by the IT Service. The second type are personal pages which belong to members of the university. All members of the university can have access to the university Web site if they have obtained a unix username and password. The IT Service also provide in house regulations for the users of the Web and some guidelines.

3.5.1 Good Points

The audience has been well targeted and future developments will seek to target the audience more specifically. The audience require and can usefully use the facilities that are provided by

the IT Service on the Web. The rules are well set out for the users to follow, there are also some guidelines which are clear and easy to follow too. None of the standards or guidelines will provide problems to the users as all the facilities are available.

Maintainability Attributes

The site had a structure for official pages which helps to improve the maintainability.

Readability and Quality Attributes

The contents of the main document are concise and easy to understand. The standards that are set up by the IT Service, if adhered to will provide more readability and better quality pages.

3.5.2 Problems

The problems broadly divide up into technical and managerial problems.

Technical Problems

The major problem with the IT Service set up is that everyone with a username and password has access to the Web, so it is difficult to monitor the quality of the site. Despite there being clear guidelines and rules for presentation and content it is difficult to monitor whether these rules are being adhered to and to enforce them. No quality controls on the contents of the document have been set up. This could mean that the standards, while they exist may not be of much use.

There are links on the Durham University pages that are out of date. On http://www.dur.ac.uk/ITService, Bibliography for the WWW should not have address http://info.cern.ch/hypertext/WWW/Bibliography.html but http://www.w3.org/hypertext/WWW/Bibliography.html. This was changed before October 1995, which is the date of the last update. This indicates no real link checking is carried out. When the page was eventually reached it transpired that it was out of date.

It is difficult at times to navigate the main university pages and no index is provided to help. This makes the site less readable. The overall structure of the site is enormous, if personal pages are included. This will often mean that problems of disorientation are encountered. The structure is expanding exponentially thus the problem will only get worse.

While analysing the log files would be useful and programs are run. Major problems from the analysis of log files cannot be reported to each user as the task would be too great.

Managerial Problems

It is debatable, whether having a smaller team to run a larger site is a good point or a problem. It can be argued that this is economic use of resources, however, effective management and maintenance would probably need the attention of more staff. In order to cope with the overhead of work, all maintenance is thus done through prompting rather than periodic maintenance checks. Thorough checking for failure to comply with the rules would again pose a very big task.

There doesn't appear to be any effort dedicated to maintenance. While the IT Service offers tools for development, it does not suggest that proper maintenance will be done and does not provide tools for maintenance.

With such a wide variety of usage of the WWW it is difficult to keep a central focus on the big picture.

Other Problems

Many of the authors lack authoring experience and poor quality of authorship.

3.5.3 Points for Consideration

It is good to bear in mind the focus of the Web, in order to carry on with the intended purpose of the site. As a site grows, it is easy to lose track of the purpose and the site loses its relevance to the audience.

It is very possible that with the large number of contributions to the site, much repetition may occur, if it hasn't already. It is probably advisable to keep a record of what sort of contributions have been made.

If changes are made to the standards, users having previously read the standards and guidelines, may not be aware that they have changed and thus not be aware they are breaking the rules. Regular communication of changes would be helpful.

Automated link checking and guidelines for individual maintenance of the pages may improve the quality of the university site overall, as well as classes on how to design Web pages, with regard to structure, quality and maintenance.

A final consideration, as with the CCTA, is that, it is not only experts in information systems that use this facility, but many others may produce Web documents and thus the quality problems are greater.

3.5.4 Suggested Improvements

Periodic maintenance checks would help improve the quality of the site, even if it is just checking for broken links. Other maintenance checks could involve, checking that pages are regularly updated. In an institution such as a university, there is a high turnover of staff and students and when a maintainer of a page leaves they may not necessarily nominate a new maintainer for the page. Periodic maintenance checks and analysis of log files could identify problems which the maintainer of a page was unaware of. Communicating these problems could be useful.

In order to promote maintenance as well as development, providing maintenance tools for the Web would be useful. These tools could be link checkers and HTML validators for example.

In order to prevent problems of corporate image, some monitoring of quality of content from pages that are more traceable, such as postgraduate pages, should be carried out. Postgraduates personal pages may well not be part of the university official Web site, but they are frequently linked to their respective departmental pages and thus can be traced as being part of the official Web site.

As the site continues to grow, an index of the content of the site may improve navigation.

3.5.5 Foreseeable Problems

There are three main foreseeable problems: size, cost and the introduction of Java. The site, if it continues to expand at the same rate, may become unmanageable, the structure growing exponentially. As the size increases, the maintenance costs of running the site may become quite high.

A further problem could be the introduction of Java applets. The capability of Java applets is not yet known and they may present serious security problems for the site.

3.5.6 A Summary of the IT Service

Like the CCTA, there are many problems, such as lack of standards for the Web, which cause the IT Service maintenance problems, which are beyond their control. However, although, a lot of thought has gone into the development and management of the site, not much thought has been given to the maintenance of Web documents on a large scale or and individual basis. The introduction of maintenance procedures could greatly overcome the problem of the increasing size of the Web site.

3.6 Conclusions from Case Studies and Major Problems Identified

This section gives an overall view of the major problems that exist and will be addressed by this research. Through the literature survey and the case studies, a number of problems were identified, they mainly fell into the following categories: lack of tools, lack of standards and guidelines, structure problems, authors lacking experience, poor maintenance procedures and problems with collaborative authoring.

The lack of tools seemed to create many problems which lead to poor maintenance. It was identified in section 2.3.5 that there is a lack of tools for development and maintenance and those which exist are limited in their capabilities. The lack of tools is more particular to maintenance than development it has to be said (2.4.3). The kind of tools that are required are tools for producing maintenance information and testing coverage (analogous to testing programs (2.5.2.8)). It also appears that in areas where tools do exist they do not exist across the range of available platforms (3.3.2). Maintenance tools are not promoted, like development tools are and thus little encouragement is given to carry out maintenance (3.5.2).

Many problems were identified as being problems of structure. Hypertext has long been known as having problems of orientation and navigation (2.3.5). However, the Web seems to have highlighted this problem with its size, the growth of structures on the Web is exponential (2.4.3.1) which leads to problems of maintenance as the complexity of the structure increases (3.3.3, 3.3.5, 3.5.2) and as the complexity increases the inclination to maintain the structure properly decreases. Unfortunately, some of the problems of structure are also due to the poor authoring of the document, in section 2.4.3 it was identified that good structures exist only by design. These are not the only problems that occur. Many authors lack experience, this problem

was identified in the literature survey (2.4.3.1) and through both case studies (3.3.2, 3.5.2). This is due to a limited understanding of the technology of hypertext (2.4.3.1) in part, as well as HTML not being a simple language to master (2.3.5). The consequences of authors having limited experience is poor quality documents, there is much over or under use of the technology, there is poor design of their visual appearance and in some cases, no design at all (2.4.2.4) which culminates in the Web being littered with poor quality documents that are difficult to maintain.

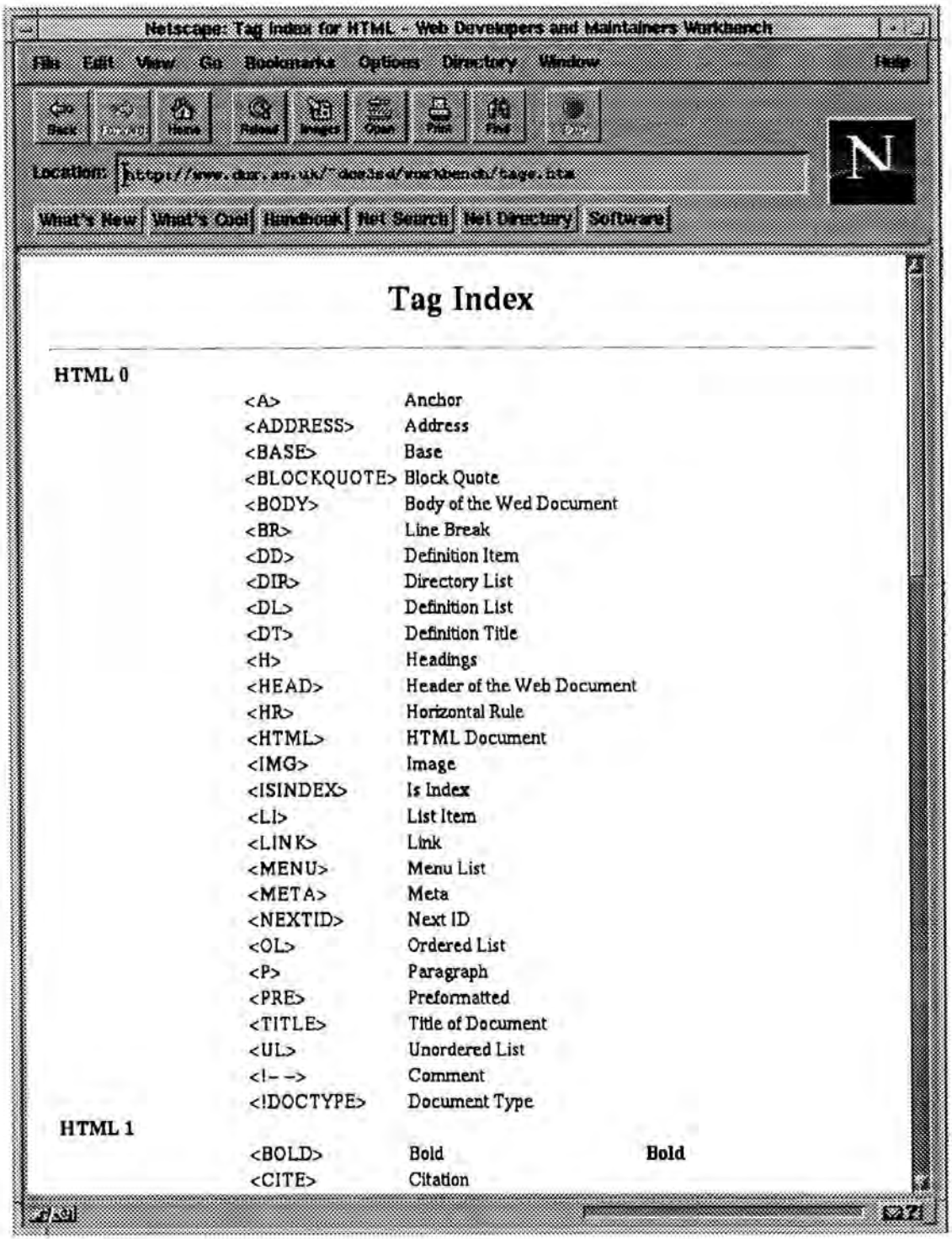


Figure 5 A browser that supports tables

Figure 6 A browser that does not support tables

A major problem that makes maintenance difficult is the lack of consensus on quality. An example of the problem is shown see *Figure 5 A browser that supports tables* & *Figure 6 A browser that does not support tables*. They both display the same page. Here it can be seen

that the quality of the document has not properly been checked as it is not supported by all browsers. Documents are of poor quality as there are no standards and guidelines (2.4.3.2) nor is professional editing practiced and thus authors are not subjected to criticism and rejection like authors of paper publications. There is also no real definition of Web maintenance to guide maintainers (2.3.5). There is a need to standardise development and maintenance (2.3.5) in order to help with the difficulty assessing quality (2.5) and producing high quality documents. At some sites, it was seen through the case studies, house standards and guidelines exist. Unfortunately, these are not always enforced (3.3.5), this is often because the overhead of work involved in checking for compliance with standards and guidelines is massive (3.5.2). Some standards can create technical difficulties for the developers and maintainers as they do not have the capabilities to comply (3.3.2). It was also identified that these standards and guidelines are not always well communicated to the developers and maintainers, particularly changes to them (3.5.3).

As there are no standards and limited guidelines for maintenance, it is difficult for many maintainers and managers to perform good maintenance. Maintenance is problematic, through the moving and deleting of files (2.4.3.1) as well as the renaming of resources (3.3.2, 2.3.3.4) which often causes broken links. Manual checking for broken links is rarely done thoroughly as the overhead in time and tedium is high (2.4.3.1) Organisations rarely perform periodic maintenance checking (3.3.2, 3.5.2), there is rarely any rigorous checking on the quality of documents (3.3.2, 3.5.2) and the overhead of work created in order to provide feedback to the maintainers of documents from the analysis of log files prevents organisations from doing this (3.3.2, 3.5.2). Organisations set up their sites in different ways, some choose to have single point of access to the site through the management, thus increasing the maintenance overhead for the management (3.3.2). Other organisations choose to allow access to the server to all maintainers and developers thus increasing the problems of quality monitoring (3.5.2). It is difficult to assess quality in order to provide good maintenance (3.3.2, 2.5.2) and as an application grows old, maintenance schedules are abandoned, hence the Web will have large sections which are deteriorating in quality.

Finally, collaborative authoring is one of the major advantages of the Web. Unfortunately, there is no sophisticated support for collaborative authoring, HTTP only really provides support for collaborative reading (2.3.5), however, documents are often written by several authors and maintained by a number of maintainers making the maintenance task all the more complex (2.4.3).

3.7 Developing a Solution

This section aims to give a short discussion in the ways in which this work could contribute towards solving some of the problems which have been identified. The major conclusion found so far is that support for maintenance is required in order to improve the quality of the Web. Two solutions appear to be the the first stage to solving the problems. The first is to isolate a specific problem and try to provide a comprehensive solution. The second is to provide some sort of object which will give the maintainer support. This could be in the form of tools, methods or guidelines.

3.8 Summary

This chapter shows that through the study of two case studies that there can be very varied approaches to the management and the maintenance of a Web site. The two case studies managed to reveal a host of other problems associated with Web maintenance. These were put together with the problems identified in chapter two to form a conclusion to the work done so far. The chapter ends with a short discussion on the way in which this work could contribute to solving some of the problems of Web maintenance. This appeared to be through isolating a specific problem and providing a solution for overall support for maintenance. In the following chapter the solution which will be provided by this work is discussed in full.

CHAPTER 4

DEVELOPMENT OF A SOLUTION

4.1 Objectives of the Chapter

This chapter aims to describe a proposed solution which will be implemented in this work. It will then give the requirements analysis and requirements specification. The requirements analysis will first be done from both case studies and user requirements and then a synthesis of the requirements will be given. A requirements specification is then given from the requirements analysis. The chapter is concluded with a section describing the design and a section on the implementation.

4.2 A Proposed Solution

From chapter two and chapter three, a proposed solution will be defined in this section, which is within the scope of this work. The major problems encountered were the lack of development and maintenanance methods and tools, the lack of standards and guidelines and the inexperience of many authors. It is of course not possible within the scope of this work to solve some of these problems comprehensively, however, support for the problem could be provided. The proposed solution is to provide a workbench that will provide some methods for development, maintenance and management and many guidelines as well as guidance on the development, maintenance and management of Web documents and Web sites.

4.3 Requirements analysis

This work has proceeded using a traditional software engineering approach. It began with a thorough requirements analysis for each case study and user requirements based on the author's experiences and problems defined in chapter two. This is followed by a synthesis of the requirements analyses.

4.3.1 CCTA Requirements

Below is a list of requirements relating to the needs of the CCTA information service. The requirements are specific to the work done at the CCTA to create a Web site that is well structured and maintained.

4.3.1.1 DOS-UNIX Problems

REQ-CCTA-01 The CCTA requires a DOS-UNIX file naming convention in order to eradicate broken links, caused by checking for errors using a DOS mirror.

4.3.1.2 Managerial Problems

REQ-CCTA-02 The CCTA requires a method for ensuring that when a number of alternate services are available, then any alteration made will not leave inconsistencies.

REQ-CCTA-03 The CCTA requires a mechanism for making an identical alteration/addition to all pages available on their server. An example of this would be changing the image for the "home" button. In order to retain uniformity, all the pages having this "button" would need to be changed.

REQ-CCTA-04 The CCTA requires a method for checking that guidelines and standards are followed.

REQ-CCTA-10 The CCTA requires a process that increases the speed with which updates are made to Web pages.

REQ-CCTA-14 The CCTA requires a process of providing feedback on problems relating to the document, to the authors/maintainers of the documents.

REQ-CCTA-18 The CCTA requires guidelines on maintaining the size of a site and thus preserving the maintainability of such a site.

REQ-CCTA-20 The CCTA requires a guideline for checking for pages that have not been updated for long periods of time.

REQ-CCTA-23 The CCTA requires checks to be made on a "live" Web too. This is done, however, not with enough regularity.

REQ-CCTA-25 The CCTA requires a process for efficient feedback of information from log files.

REQ-CCTA-24 The CCTA requires a process for communicating changes in guidelines to organisations who have become independent.

4.3.1 CCTA Requirements

Below is a list of requirements relating to the needs of the CCTA information service. The requirements are specific to the work done at the CCTA to create a Web site that is well structured and maintained.

4.3.1.1 DOS-UNIX Problems

REQ-CCTA-01 The CCTA requires a DOS-UNIX file naming convention in order to eradicate broken links, caused by checking for errors using a DOS mirror.

4.3.1.2 Managerial Problems

REQ-CCTA-02 The CCTA requires a method for ensuring that when a number of alternate services are available, then any alteration made will not leave inconsistencies.

REQ-CCTA-03 The CCTA requires a mechanism for making an identical alteration/addition to all pages available on their server. An example of this would be changing the image for the "home" button. In order to retain uniformity, all the pages having this "button" would need to be changed.

REQ-CCTA-04 The CCTA requires a method for checking that guidelines and standards are followed.

REQ-CCTA-10 The CCTA requires a process that increases the speed with which updates are made to Web pages.

REQ-CCTA-14 The CCTA requires a process of providing feedback on problems relating to the document, to the authors/maintainers of the documents.

REQ-CCTA-18 The CCTA requires guidelines on maintaining the size of a site and thus preserving the maintainability of such a site.

REQ-CCTA-20 The CCTA requires a guideline for checking for pages that have not been updated for long periods of time.

REQ-CCTA-23 The CCTA requires checks to be made on a "live" Web too. This is done, however, not with enough regularity.

REQ-CCTA-25 The CCTA requires a process for efficient feedback of information from log files.

REQ-CCTA-24 The CCTA requires a process for communicating changes in guidelines to organisations who have become independent.

4.3.1.3 Technical Problems

REQ-CCTA-05 The CCTA requires guidelines that do not require the use of specific features only available to specific browsers. An example of the problem would be when requesting that pages be centred, centring of pages is a Netscape feature that does not appear on browsers such as Mosaic.

REQ-CCTA-06 The CCTA requires guidelines that are not specific to hardware such as the size of the screen.

REQ-CCTA-07 The CCTA requires guidelines that guide the developer in developing pages that have a good appearance on all screens.

REQ-CCTA-08 The CCTA requires guidelines to help developers title their documents properly.

4.3.1.4 Development

REQ-CCTA-15 The CCTA requires guidelines for focusing developers on the purpose of their Web pages.

REQ-CCTA-16 The CCTA requires guidelines on preserving the relevance of Web pages.

REQ-CCTA-17 The CCTA requires guidelines to help developers avoid repetition.

REQ-CCTA-19 The CCTA requires guidelines on removing pages from a set of Web pages.

REQ-CCTA-26 The CCTA requires a teaching method for design of Web pages.

REQ-CCTA-31 The CCTA requires a design method.

REQ-CCTA-33 The CCTA requires a workbench geared towards non-professional developers.

REQ-CCTA-12 The CCTA requires teaching support.

4.3.1.5 Maintenance Problems

REQ-CCTA-09 The CCTA requires a process which includes periodic checking for maintenance problems.

REQ-CCTA-11 The CCTA requires an in-depth process for checking the quality of the document.

REQ-CCTA-13 The CCTA requires more sophisticated tools for maintenance. These would help improve the maintenance. The current tool used by the CCTA performs some maintenance tasks (see appendix B).

REQ-CCTA-27 The CCTA requires a method for tracking problems that have been dealt with or that are likely to reappear in the future.

REQ-CCTA-28 The CCTA requires a mechanism for getting permission to make changes to a page.

REQ-CCTA-29 The CCTA requires a mechanism for the automated checking of pages before putting them on the Web.

REQ-CCTA-30 The CCTA requires a mechanism for monitoring quality of the content of documents.

REQ-CCTA-32 The CCTA requires more use of tools for various activities of maintenance.

4.3.2 ITS Requirements

The IT Service requirements are considerably fewer in number than those of the CCTA. This is because the requirements stated below are less specific than those of the CCTA. The reason for the different style of requirement is due to the different management techniques of the organisations. The fundamental difference is that CCTA do not give direct access of the World-Wide Web server to their users. Instead the users submit Web documents to the CCTA which are then monitored and placed on the Web server. This approach significantly reduces problems of monitoring quality at the Web site. In the IT Service case, all users have access to the Web site.

4.3.2.1 Management

REQ-ITS-01 The ITS require a mechanism for controlling the quality of document contents.

REQ-ITS-02 The ITS require a mechanism for enforcing standards.

REQ-ITS-03 The ITS require automated checking for compliance with regulations.

REQ-ITS-08 The ITS require a mechanism for providing feedback from errors detected in the logfiles.

REQ-ITS-09 The ITS require a mechanism for checking pages that have not been updated in a long time and to alert the maintainer.

REQ-ITS-10 The ITS require a system for effective communication of changes in standards and guidelines to the developers and maintainers.

4.3.2.2 Development

REQ-ITS-04 The ITS require better structure for navigating the site.

4.3.2.3 Maintenance

REQ-ITS-05 The ITS require a process for periodic link checking for broken links.

REQ-ITS-06 The ITS require a process for overall periodic maintenance checks.

REQ-ITS-07 The ITS require tools for maintenance.

REQ-ITS-11 The ITS require a process for quality checking.

REQ-ITS-12 The ITS require a method for keeping track of problems that arise regularly.

4.3.3 Own Requirements

The requirements listed in this section are based on the needs of the developer. These needs have been touched upon in the two previous sections but the focus then was from a managerial angle.

4.3.3.1 Development

REQ-OWN-01 A process model for development that will produce good quality documents that are readable and maintainable. The process model will include design and development procedures.

REQ-OWN-02 Guidelines on the content of a Web page which help the developer define a purpose for the page, target the correct audience and make the pages concise and informative with correct usage of graphics.

REQ-OWN-03 Guidelines on developing pages that are easy to navigate and do not render the user disorientated.

REQ-OWN-04 Guidelines on how to structure Web pages.

REQ-OWN-05 Facilities for improving authoring abilities.

REQ-OWN-06 Supporting development for maintenance - good structuring
- nominating a maintainer
- documentation
- ensuring information preserves its relevance
- using feedback

REQ-OWN-13 All standards need to be complied with when developing a Web page and in as far as possible, all guidelines should be adhered too. Hence, a mechanism is required in order to ensure this is done.

REQ-OWN-14 A design method is required.

REQ-OWN-16 Guidelines on titling documents and usage of buttons in order to make the document more navigable are required.

4.3.3.2 Maintenance

REQ-OWN-07 A mechanism for preserving the structure of a Web document
- links
- graphics

- ftps
- mailto facilities
- files
- removing pages

REQ-OWN-08 A process for checking the quality of Web documents

REQ-OWN-09 Version control

REQ-OWN-10 Guidelines for maintenance of Java applets.

REQ-OWN-11 Guidelines and tools for navigation and visualisation of Web documents.

REQ-OWN-12 When an alteration is made to one page, changing a feature that appears on all other pages in the document, some sort of mechanism is needed to ensure that all the appropriate changes are made.

REQ-OWN-15 Where possible automated checking for problems is required.

4.4 Synthesis of Requirements

This section gives a synthesis of the previous requirements.

4.4.1 Requirements for Management

REQ-SYN-MAN-01 There is a requirement for a method which ensures alterations are made without leaving inconsistencies in the Web document. The following cases should not arise:

- A number of platform dependent alternate services are offered, any change to one without changing the others will leave inconsistencies. (REQ-CCTA-02, REQ-OWN-09)
- A feature appearing on all pages may require changing, this could be a button or logo, some pages may not be altered. (REQ-CCTA-03, REQ-OWN-12)

REQ-SYN-MAN-02 There is a requirement for a process that will provide feedback from errors that have been tracked down by the error logfiles of the server (REQ-CCTA-14, REQ-ITS-08) and other problems relating to the document which have been tracked through other forms of feedback such as the complaints of users. (REQ-CCTA-25)

REQ-SYN-MAN-03 There is a requirement for a process which will make updating pages more efficient. (REQ-CCTA-10)

REQ-SYN-MAN-04 There is a requirement for a method of maintaining the size of a site thus preserving the readability, quality and maintainability of the site.

REQ-SYN-MAN-05 There is a requirement for guidelines for the management to track down pages which have not been updated for a long period of time and alert the maintainer of this fact. (REQ-ITS-09, REQ-CCTA-20)

REQ-SYN-MAN-06 There is a requirements, that checks to see whether a Web page(s) are "ok", i.e. to the management or maintainer's satisfaction, should be made on a "live" Web (REQ-CCTA-23). This would also eradicate the problem occurring form DOS-UNIX conversions (REQ-CCTA-01).

REQ-SYN-MAN-07 There is a requirement for a set of guidelines on quality control of documents. (REQ-ITS-01)

REQ-SYN-MAN-08 There is a requirement for better quality control of document content. (REQ-ITS-01)

REQ-SYN-MAN-GUIDELINES

There are a number of requirements concerning the use of guidelines in the management of a Web site.

REQ-SYN-MAN-GUID-01 There is a requirement to ensure that standards are adhered to. (REQ-ITS-02)

REQ-SYN-MAN-GUID-02 There is a requirement for enforcing standards to thus preserve quality. (REQ-ITS-02)

REQ-SYN-MAN-GUID-03 There is a requirement for guidance with checking that guidelines and standards are followed. (REQ-CCTA-04, REQ-ITS-03)

REQ-SYN-MAN-GUID-04 There is a requirement for effective communication of change in standards and guidelines.

REQ-SYN-MAN-GUID-05 There is a requirement that all guidelines be hardware, platform and browser independent. (REQ-CCTA-05, REQ-CCTA-06)

REQ-SYN-MAN-GUID-06 There is a requirement that all standards and guidelines be obtained before development and be adhered to during development and maintenance. (REQ-OWN-13)

4.4.2 Requirements for Development

REQ-SYN-DEV-01 There is a requirement for facilities for improving authoring abilities of developers. (REQ-CCTA-12, REQ-OWN-15)

REQ-SYN-DEV-02 There is a requirement for guidelines for:

a) focusing developers on the purpose of the WWW document. (REQ-CCTA-15)
b) preserving the relevance of the WWW document (REQ-CCTA-16)
c) removing pages from the WWW document/site (REQ-CCTA-17)
d) avoiding repetition (REQ-CCTA-19)
e) structuring the Web site in order to have easy to navigate pages (REQ-ITS-04, REQ-OWN-04, REQ-OWN-03, REQ-CCTA-22)
f) obtaining and adhering to standards and guidelines (REQ-OWN-13)
g) for targeting the audience (REQ-OWN-02)
h) guidelines for quality (REQ-OWN-01)

REQ-SYN-DEV-03 There is a requirement for the support for maintenance through:

a) good structuring of the pages
b) nominating a maintainer of the pages
c) documentation of the development and maintenance
d) ensuring that the information presented will preserve its relevance
e) setting up a feedback mechanism. (REQ-OWN-06)

REQ-SYN-DEV-04 There is a requirement for a design method. (REQ-CCTA-31, REQ-OWN-14)

REQ-SYN-DEV-05 Guidelines on titling documents and usage of buttons in order to make the document more navigable are required. (REQ-OWN-16)

4.4.3 Requirements for Maintenance

REQ-SYN-MAINT-01 There is a requirement for a process that incorporates periodic maintenance checking. (REQ-CCTA-09, REQ-ITS-05, REQ-ITS-06)

REQ-SYN-MAINT-02 There is a requirement for the increased use and availability of maintenance tools. (REQ-ITS-07, REQ-CCTA-13, REQ-CCTA-32)

REQ-SYN-MAINT-03 There is a requirement for a set of guidelines or a process model for quality checking. (REQ-CCTA-11, REQ-CCTA-30, REQ-OWN-08, REQ-ITS-11)

REQ-SYN-MAINT-04 There is a requirement that the management or the maintainer track frequents problems and stop them recurring. (REQ-CCTA-27, REQ-ITS-12)

REQ-SYN-MAINT-05 There is a requirement for where possible, the automated checking of Web pages. (REQ-CCTA-29, REQ-OWN-15)

REQ-SYN-MAINT-06 There is a requirement for a mechanism for preserving the structure, this entails checking the following:

- links
- graphics
- ftps
- mailto facilities
- files (REQ-OWN-07)

REQ-SYN-MAINT-07 There is a requirement for guidelines for the maintenance for Java applets. (REQ-OWN-10)

REQ-SYN-MAINT-08 There is a requirement for guidelines for the navigation and visualisation of Web pages. (REQ-OWN-11)

4.5 Requirements Specification

This section follows the software engineering approach giving a requirements specification. This is split into an introductory section and a general description. As the workbench is not a specific piece of software, some parts that are usually included in a requirements specification have been omitted. This is for example the section on input/output and processing requirements.

4.5.1 Introduction

The introductory section defines the purpose and gives an overview of the workbench.

4.5.1.1 Purpose

The purpose of the workbench is primarily to provide the appropriate help to developers and maintainers of World-Wide Web documents. However, it is also intended to help those who manage and host Web sites with regard to the development and maintenance of Web documents, for example, producing guidelines and standards for within the site. The purpose is to guide developers in producing good quality documents, that are easily maintainable. It intends to provide the appropriate tool kit and procedures where possible for the development and maintenance of Web documents. The workbench will also help developers and maintainers produce easy to navigate documents which will alleviate the problems that exist with the development and maintenance of hypertext documents.

4.5.1.2 Scope

The workbench is being produced as part of a Master's degree, so all the work done for this has to take place within the 12 months with the equipment provided.

4.5.1.3 Overview

The workbench will be a World-Wide Web based application. This will allow it, in as far as possible, to be platform independent. The workbench will be split into the following sections: development maintenance and management. Each section will be in the form of a World-Wide Web page. The page will contain methods, guidelines and tools and will be laid out in order to lead the developer or maintainer through the sections in the correct order.

4.5.2 General Description

This section gives a general description of the workbench. It is divided up into a development, maintenance and management.

4.5.2.1 Development

This section of the workbench is aimed at the development of Web pages. This is important for the maintenance side of the Web, as it can help reduce the maintenance load if the pages are well developed. It aims to help developers in developing good quality, readable and maintainable pages by giving design and development procedures supplemented by the appropriate rules and guidelines. (REQ-SYN-DEV-02(f), 04)

4.5.2.1.1 Content

This sub-section will be devoted entirely to the content of a Web page. Within the requirements analysis, a need for aid with regard to the content of a Web page was found. From this it was decided that the workbench would provide guidelines for:

a) focusing developers on the purpose of the WWW document
b) preserving the relevance of the WWW document
c) avoiding repetition
d) obtaining and adhering to standards and guidelines
e) targeting the audience
f) quality

(REQ-SYN-DEV-02)

4.5.2.1.2 Navigation

One of the major problems associated with hypertext, is the problem of disorientation. It is often difficult to navigate through documents in the correct order. You can often arrive in the middle of a document and it becomes difficult to establish the context. It is for this reason, that aid with proper titling of documents, should be addressed. Advice on buttons and indexes should also be given. (REQ-SYN-DEV-05)

4.5.2.1.3 Structure

A factor in the readability, navigation, development and maintenance of hypertext is the structure. In order to make a good readable document, a good structure is required for the document. For the document to be easy to maintain, it must be easy to append more links to the structure whilst preserving the overall structure. This section will concentrate on how to develop good structures. (REQ-SYN-DEV-03(a))

4.5.2.1.4 Learning HTML/Java/VRML

A problem noted by the CCTA and many other sources[13], [12], is that there is a lack of authoring experience amongst the development community. The people that are doing the job of creating documents on the Web invariably have little experience in the use of HTML, creating hypertext documents or other features such as Java and VRML. This section aims to provide the appropriate learning materials for HTML, Java and VRML. This will be accompanied by information on the appearance of pages across different platforms, advice on the different versions of HTML and a glossary of terms.(REQ-SYN-DEV-01)

4.5.2.1.5 Developing for maintenance

In this section, a set of guidelines which can be applied during development, will be put forward in order to facilitate the maintenance process. These will be: good structure, nominating a maintainer, providing maintenance information, documentation, making sure that the information presented preserves its relevance and the use of feedback. (REQ-SYN-DEV-03)

4.5.2.2 Maintenance

This section concentrates on all the maintenance activities that the workbench will cover. It will primarily provide the user with an approach to maintenance and will incorporate the major activities that should be undergone during the maintenance of Web documents (REQ-SYN-MAINT-01, REQ-SYN-MAN-03). The activities that are incorporated in the process model will also require aids in the form of tools and guidelines for the user.

4.5.2.2.1 Structure Maintenance

The problems that arise when trying to preserve the structure are the following:

- Broken links
- Graphics that are no longer connected or do not display properly.
- Ftps sites that have been moved or do not contain the relevant documents/software any longer.
- Mailtos with incorrect e-mail addresses.

(REQ-SYN-MAINT-06).

A major problem discovered, is the overhead of work involved in maintenance, to alleviate some of the task, tools are required. In as far as possible, tools to automate maintenance activities will be included (REQ-SYN-MAINT-02, 05).

4.5.2.2.2 Quality Checking

A second factor of maintaining Web documents, is detecting problems with the content of a Web page. This can range from mistakes that need correcting, to poor quality of the document content or structure, that should be brought up to standard. Of course, it is difficult to say what is and isn't a good quality document, so here we provide merely guidelines and approaches to improving quality. The kind of problem involved would be:

- Page content out of date.

- Targeting the right audience.
- Grammar and spelling mistakes.
- Making sure the information presented preserves its relevance.
- Making sure documents don't become uncontrollably large.
- Using feedback. (REQ-SYN-MAINT-03, REQ-SYN-MAN-08)

4.5.2.2.3 Maintenance Documentation

Outlines of the kind of documentation and how it can be put to good use are discussed in this section of the workbench. Some documentation can be generated automatically, other documentation will have to written manually.

The documentation which will be discussed will be:

- maintenance documentation
- feedback
- documentation of the development and change
- statistics
- documentation of frequent problems in order to predict and prevent their recurrence. (REQ-SYN-MAINT-04)

4.5.2.2.4 Version Control

A problem discovered by the CCTA was that inevitably certain services offered would be platform dependent. Thus an effort was made to accommodate most platforms. To do this, an alternative service was provided for each platform. However the problem arises when alterations were made to one of the services. The services became inconsistent and thus a mechanism or guidance is required by the workbench to solve this problem (REQ-SYN-MAN-01).

4.5.2.2.5 Java Maintenance Problems

The CCTA and the IT Service recognised that Java is rapidly growing and Java applets will soon be common place on Web pages. It is fair to assume that such applications will have the same problems as normal software applications. As the developers of Web pages tend not to be software professionals, there is a strong chance that neither will the developers of Java applets be software professionals and so producing applets that need a high level of maintenance. (REQ-SYN-MAINT-07) Hence, this section will provide support for Java applets.

4.5.2.2.6 Removing Pages

One of the big problems of developing and maintaining Web pages is how to phase them out. This is the problem of removing pages and thus leaving other people with broken links. Currently developers adopt the attitude that pages, once placed on the Web should remain there in order to solve this problem. However this is not an optimal solution, when pages are restructured pages need to be moved and when pages are no longer of any use they should be removed. The WWW is growing at such a rapid rate that leaving pages where they will only increase the size more. In this section a method of removing pages and some guidelines will be given.

4.5.2.2.7 Navigation and Visualisation

A recognised problem of hypertext, is that of disorientation. When reading something that does not have a linear structure, it is easy to become disoriented. This problem was already recognised as far back as [2] (see section 2.2.5). The problem increases with the size of the structure and with the lack of design when developing hypertext applications. Web pages form part of a very large structure and are usually developed without proper design procedures. This means that pages are often difficult to navigate. This section will provide tools for visualising the structure (REQ-SYN-MAINT-08).

4.5.2.5 Management

Given the very nature of the set up of a WWW site, a large server is required which will usually host many Web pages. In order to provide some discipline to the World-Wide Web, the management of the service must be responsible for monitoring the quality of its Web pages. In conjunction with the CCTA and the Durham University IT Service, the role of such management has been identified. The following points will be addressed in this section.

- How to provide (and enforce) rules and guidelines for developers and maintainers (REQ-SYN-MAN-GUIDELINES). These will be a set of guidelines on ensuring that standards are adhered to, enforcing standards to preserve quality, effective communication of changes in guidelines, producing guidelines that are not platform, hardware or browser dependent and facilitating obtaining guidelines.
- Monitoring quality (REQ-SYN-MAN-08, 07)
- Providing feedback on logfile errors (REQ-SYN-MAN-02)
- Monitoring size and statistics of a site (REQ-SYN-MAN-04)

- A method for tracking pages that have not been updated for a long time and contacting the maintainer (REQ-SYN-MAN-05)

4.6 Design

The reason for incorporating support for development, despite the focus of the work being on maintenance, is that many of the maintenance problems were thought to occur because of poor quality of authorship.

The workbench will be developed through the requirements of the case studies, as well as through the requirements which are developed from a user angle, based on my own experience and documented problems identified in the literature survey. This has been done using the requirements synthesis. The workbench places more emphasis on support through guidelines than on the provision and use of tools. Through the case studies and literature survey, it became apparent that with the availability of some tools for development and maintenance, this does not necessarily lead to the usage of the tools. It was felt that primarily users require guidance. Some tools that are easy to use are provided with the workbench.

A second design decision was to make the workbench a Web based service, i.e. the workbench would come in the form of a hypertext application, which would appear on the Web. This would make the workbench easily accessible and in a format that most of the users would already be familiar with. The workbench is also meant as a reference guide for developers and maintainers, so when they are developing and maintaining their documents, they will undoubtedly use a browser and thus, having a Web based browser would minimise the use of resources.

In order to follow the requirements, the workbench is split into three categories, development, maintenance and management. Each category is directed at the developer, maintainer and manager respectively. Each section is then subdivided further. The development section splits into design, content, navigation, structure, learning material for HTML, Java and VRML and development for maintenance. The maintenance section splits into a model for maintenance, structure maintenance, quality checking, maintenance documentation, making changes, java maintenance, removing pages and visualisation. The management section splits into sections on providing standards and guidelines, monitoring quality, providing feedback and tracking recurrent problems.

The design of the structure was done using some of the methodolgy presented in [16]. As the Relationship Management Methodology (RMM) is geared towards very large documents, rather than a single application, it was felt that following it precisely was inappropriate, creating a far greater overhead of work, however, some of the structuring came from the RMM methodology, in particular, the Relationship Management Design Model (RMDM) access constructs. In as far as possible, an indexed guided tour was used. A diagram of this is shown in *Figure 7 An indexed guided tour.*

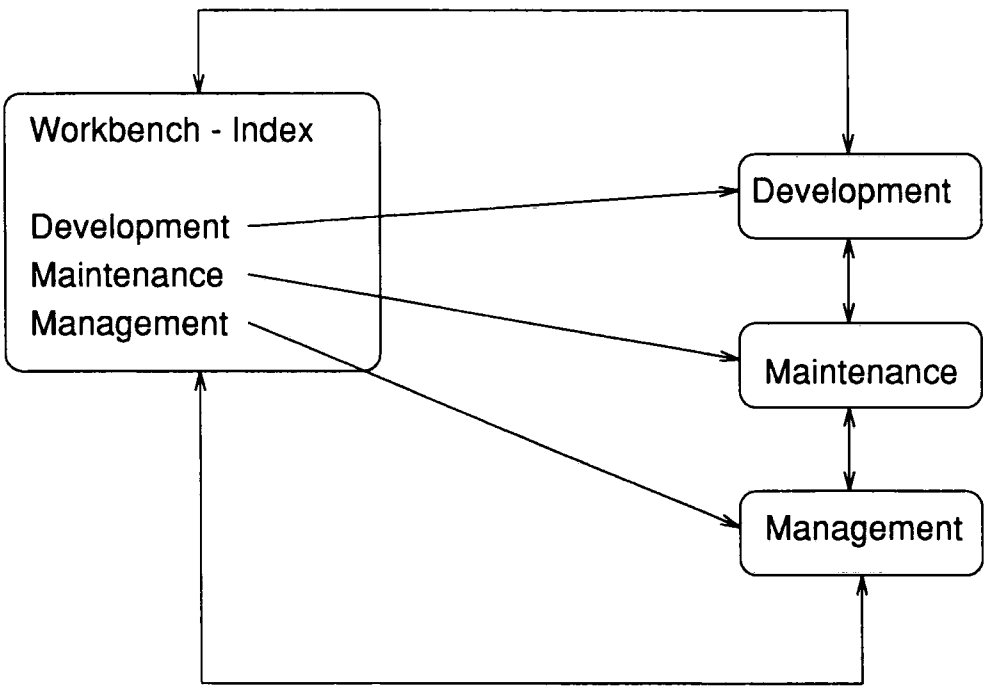


Figure 7 An indexed guided tour

This would allow users to both read the document from start to finish and use the document for reference only, allowing quick access to the intended part.

The development section is structured as shown, see *Figure 8 The structure of the development section.*

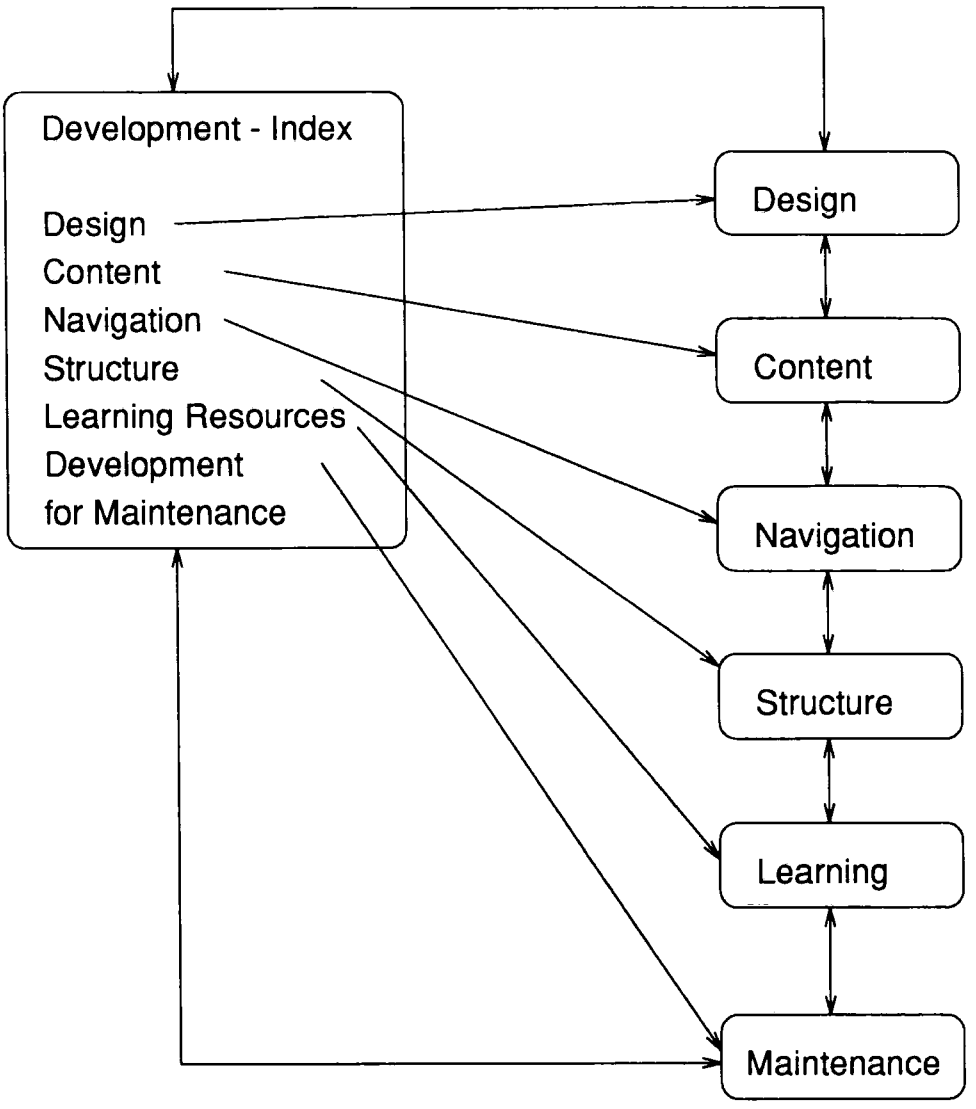


Figure 8 The structure of the development section

The maintenance section is structured as shown *Figure 9 The structure of the maintenance section.*

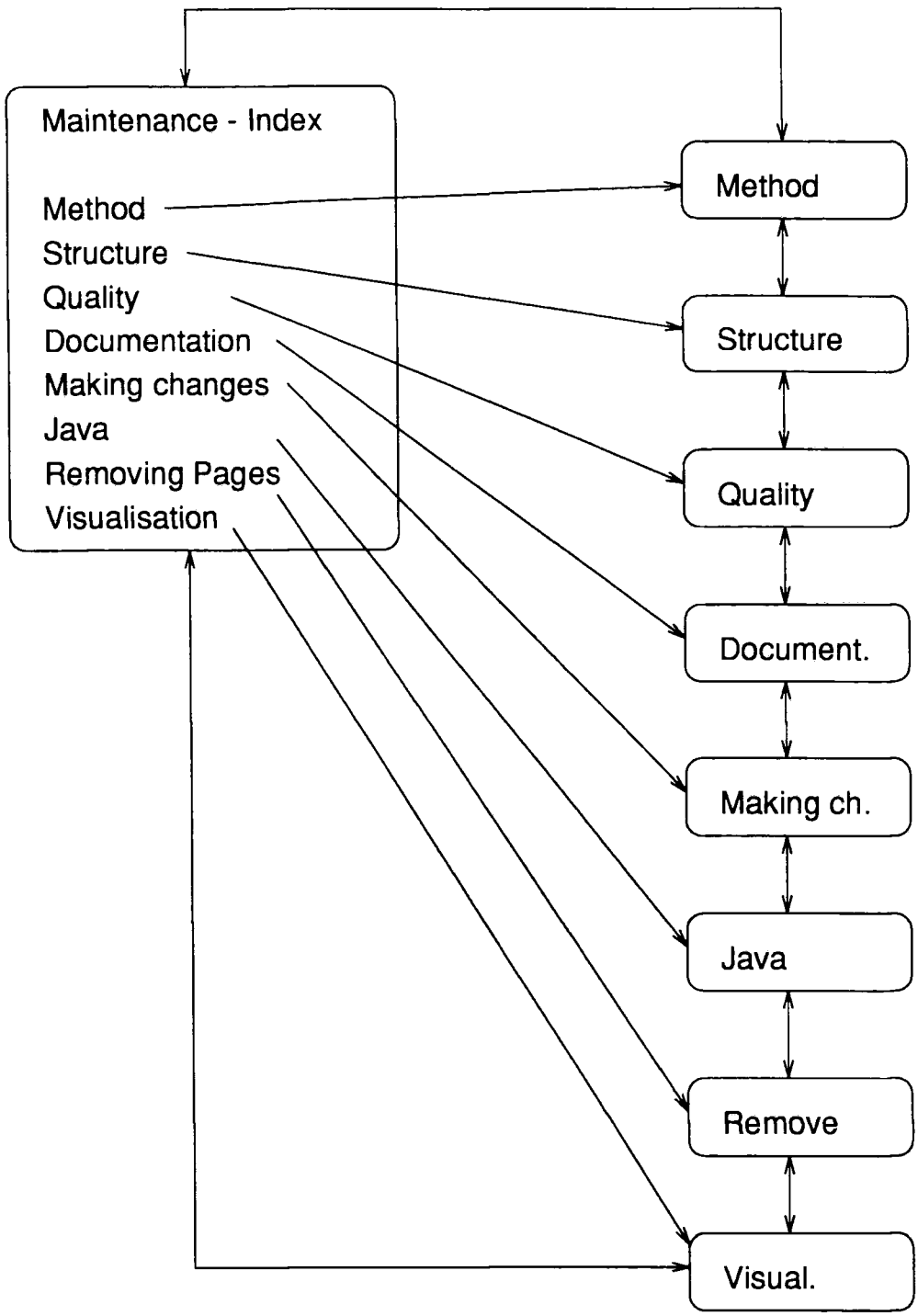


Figure 9 The structure of the maintenance section

The management section is structured as shown in *Figure 10 The structure of the management section.*

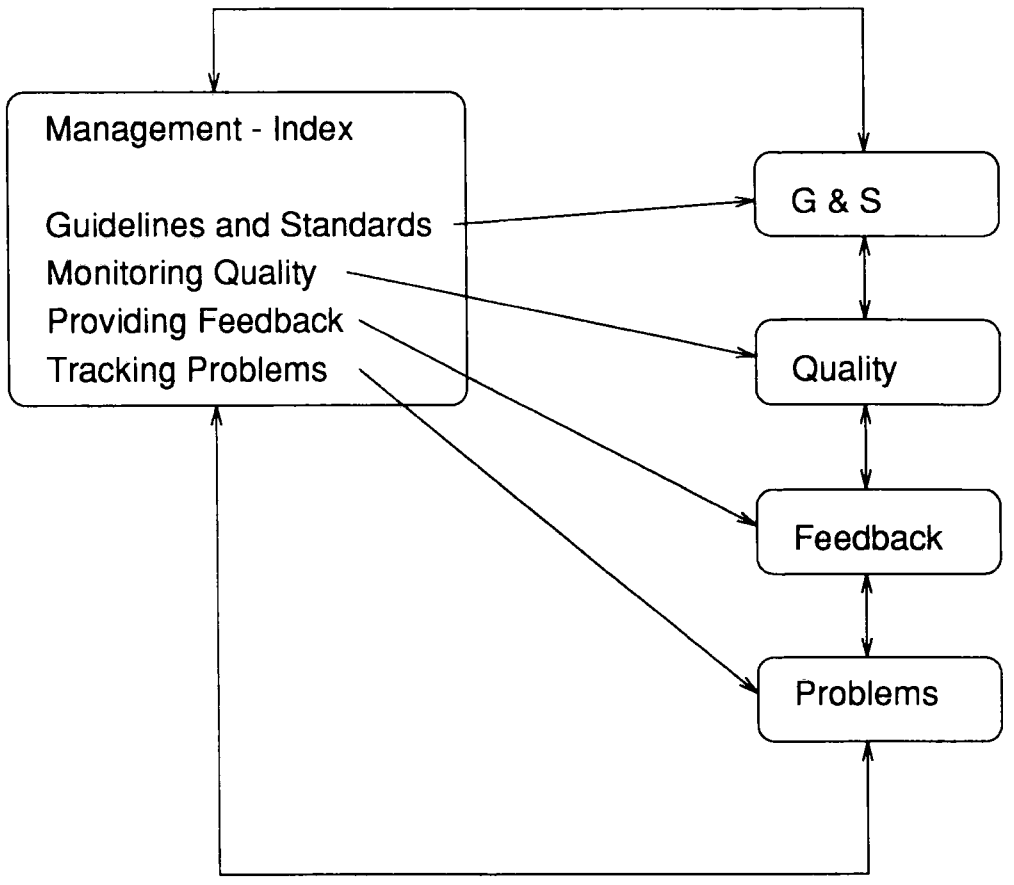


Figure 10 The structure of the management section

4.7 Implementation

For the development a PC Pentium P75 running Windows 3.1 was used. The software used was Ken Nesbitt's trial evaluation WebEdit and the beta version of Netscape Navigator Gold 3.0. The URL address of the workbench is:

http://www.dur.ac.uk/~dcs3sd/workbench/index.htm

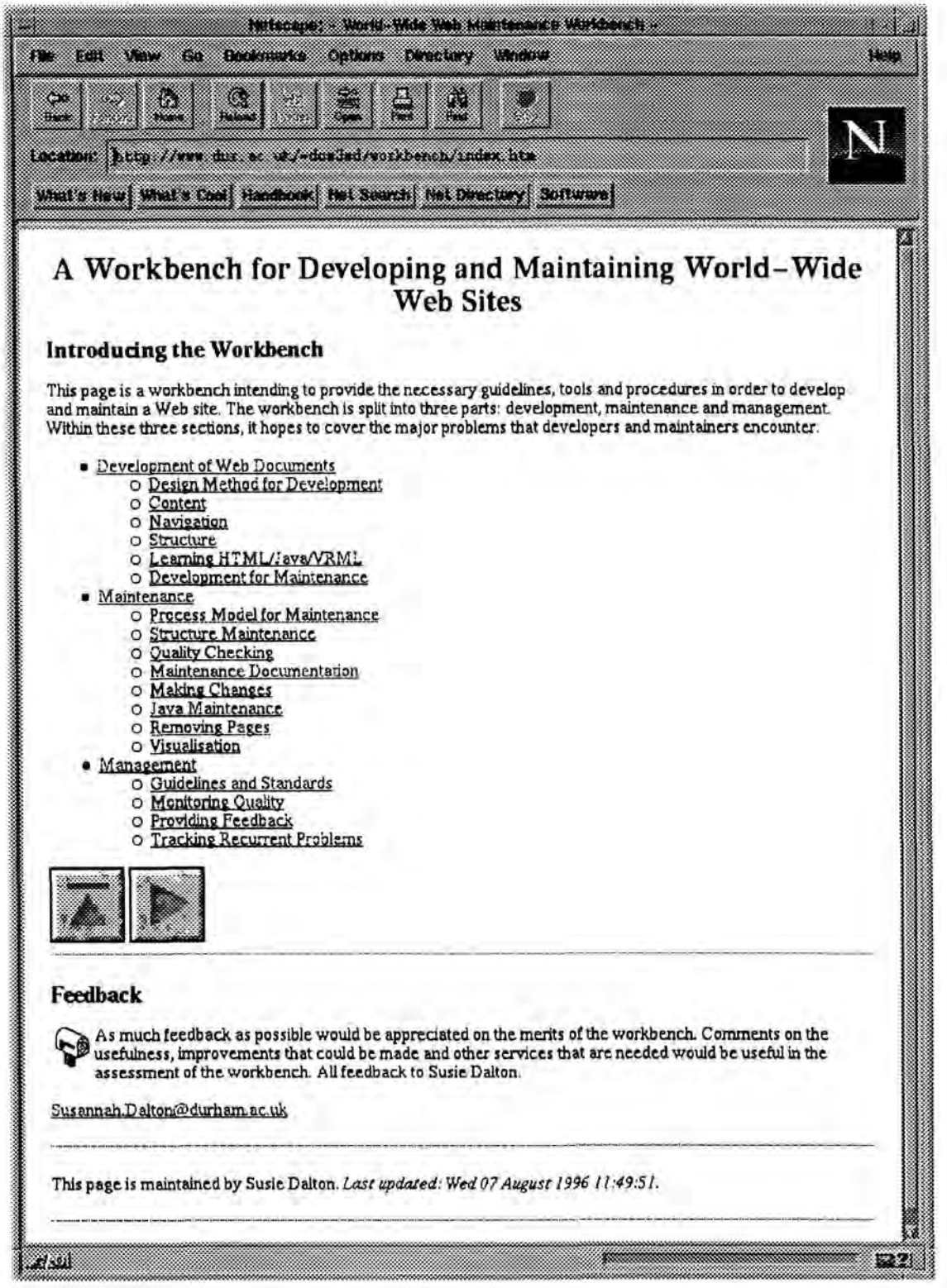


Figure 11 A screenshot of the workbench

In *Figure 11 A screenshot of the workbench*, the screen layout of the workbench is shown. It can be seen in the diagram that arrows were used at the bottom of the page in order to implement the guided tour section of the design. The upward arrow leads back to the start of

the workbench and the forward arrow leads to the next page. The diagram also indicates that the index part of the design, allowing the user to access a specific part of the workbench.

A feed back section was also implemented. This can also be seen in figure 4.3. This is done by having a small section at the bottom of the page requesting feedback information which shows the user that feedback is welcomed and the e-mail address which the user must use in order to send feedback. With certain browsers, this e-mail address is also an anchor, which when clicked upon will lead the user to a mailer.

The workbench being a Web-based service, is can be maintained by applying it to itself. It will be maintained by the author. This overcomes the problem of having to obtain new versions. The workbench also will have the advantage of not having to be obtained and installed and is platform independent. However, there are some disadvantages, some of the tools that are provided with the workbench will not be Web-based applications and so will be platform dependent. A disadvantage, will be that network problems can make other servers unreachable and servers can go down making the service unavailable at times.

4.8 Summary

This chapter describe a proposed solution to some of the problems identified in the two previous chapters. This solution is in the form of a workbench which gives WWW users methods, guidelines and tools for the development, maintenance of Web pages and management of a Web site.

CHAPTER 5

RESULTS AND TRIAL USAGE OF WORKBENCH

5.1 Objectives of the Chapter

The objectives of this chapter are to check that the workbench meets its requirements set out in chapter four and that the workbench is easy to use and provides the necessary support for development, maintenance and management of Web documents.

5.2 Does the Workbench Conform to the Original Requirements?

In order to assess the overall value of the workbench to the users, it is important to see whether the workbench conforms to its original requirements.

5.2.1 Requirements for Management

Overall, many of the management requirements are dealt with in the maintenance section. A number of the requirements are not comprehensively satisfied. Below, in more detail, a description of the discrepancies is given.

REQ-SYN-MAN-01 requires a method for dealing with inconsistencies, however, instead of providing the management with a method, the maintenance section has a section on making changes which provides guidelines for not producing inconsistencies.

REQ-SYN-MAN-03 requires a process for making updating pages more efficient. A method is given in the maintenance section of the workbench, that shows how the maintenance process could be carried out. Whether, this proposed method makes the whole process more efficient can only be determined through trial usage of the workbench. This problem could also be solved through better management structures, i.e. the number of people and the amount of resources used for the procedure. This sort of research is beyond the scope of the this work. All other requirements for management are satisfied to some extent by the workbench.

5.2.2 Requirements for Development

All the development requirements are addressed by the workbench. In some cases, however, a tool to assist a requirement was required and only a guideline was given. This is because, a tool for such purposes, did not exist or could not be obtained. Developing tools for such purposes would have been beyond the scope of the work. REQ-SYN-DEV-04 requires a design method. There are a number of design methods which have been developed. At the start of this work, however, there were relatively few. It was therefore not possible to survey all these methods properly in the time allowed, to produce the best method, instead, one method was used which was thought to be suitable.

5.2.3 Requirements for Maintenance

Not all the requirements were addressed satisfactorily in this section. REQ-SYN-MAINT-07 requires guidelines for the maintenance of Java. At the time of writing, not enough is known about Java to produce specific guidelines on the subject. Hence, some help in obtaining information on Java was given instead. Much more research is required into the maintenance problems and solutions associated with Java, in order to produce a suitable set of guidelines. REQ-SYN-MAINT-02 requires the increased use and availability of maintenance tools. This requirement could never be satisfied completely. However, tools are provided with the workbench for maintenance. The use of these tools is also encouraged through the guidelines given. Thus, the requirement has been well addressed. REQ-SYN-MAINT-03, requires either a set of guidelines or a process model for quality checking, a set of guidelines has been provided and quality checking is part of the model for maintenance. However, more research would be required to produce a process model for quality checking. In REQ-SYN-MAINT-04 a mechanism is required, however, such a mechanism for tracking frequent problems and stopping them recurring would take more time than was allowed, hence a guideline was given. REQ-SYN-MAINT-05 requires "automated checking" of Web pages. Tools for checking Web pages were provided, other checks for Web pages were suggested through guidelines. Fully automated checking is not given.

5.2.4 Result

To conclude, the requirements, in as far as possible, given the scope of the work and the time constraints, were all met. Some of requirements were addressed more satisfactorily than others and some requirements were impossible to meet, establishing that much further research would be required in order to produce a workbench that would solve the problems of development, maintenance and management.

5.3 Trial Usage

The trial usage of the workbench was done. This was carried out by going through each section and using it with a set of pages belonging to the author. Overall the workbench was thought to be user-friendly. It is not difficult to use as it does not need installing and maintaining as it is a Web based service. It is also platform independent. The pages were easy to navigate too, overcoming much of the problem of user disorientation. Unfortunately, as the service is Web based, it is difficult to use in conjunction with Web pages. This problem is apparent when trying to cross check guidelines provided by the workbench, against pages that are being developed or maintained. A printable copy of the workbench should be made available to alleviate this problem.

5.3.1 Development Section of Workbench

The design proposed in the workbench produces good Web pages. This design does, however, pose some problems. To begin with it is still complex to follow. The kind of effort it requires to follow the design method would imply that the Web pages would need to be numerous. This would mean that the document would be greater than ten pages for example. The design method also presumes that the user will have a good sound knowledge of computer science design methods, such as entity-relationship diagrams. The workbench should cater for all users and not just those knowledgeable in computer science. The method also posed the problem that it concentrated mainly on structuring the document, which, though very important is not the only factor.

In the development section, a part that proved particularly useful, was the inclusion of a quick reference guide for HTML. This was composed of a sample document, a list of the tags and a list of special characters. Most texts on developing Web documents do not include an easy to use reference guide for use in conjunction with development.

The section on developing Java and VRML applications was not in depth at all. This was because it was beyond the scope of the work to develop such a section in depth. Instead pointers to sites that give information on Java and VRML were given.

The section on development for maintenance is an important section that provides useful ideas that are in a lot of cases overlooked and certainly during the trial usage proved to be missing

from the pages used. Some of the guidelines may seem obvious and they did prove to be clear, though support for their implementation would be useful.

5.3.2 Maintenance Section of the Workbench

A method for carrying out maintenance is provided. More research would be required in order to assess whether this method was suitable enough and how the method could be improved. Despite these points, the method is a good start to maintenance. Maintenance has been shown to be an area that is often overlooked and this method provides a good starting point for those who do not maintain their pages at all. The method is also straight forward to follow and would not confuse less knowledgeable users. Support for each part of the method is provided by the workbench.

Tools have been provided in the maintenance section, particularly as an aid in the analysis section. It would be useful if more tools were provided. The ones that are provided are for the most part easy to use and give much useful information. More work on providing tools for analysis and maintenance is required in this area, but is beyond the scope of the work.

Maintenance documentation is another area, frequently overlooked by maintainers of Web documents. This area has been introduced by the workbench and is useful as a starting point for maintainers. More work is required to produce automatic document generators which would encourage the use of documentation and reduce the overhead of work.

Java maintenance is mentioned in the workbench. Java, is still relatively new to the Web and work to produce good maintenance guidelines would require much research. This area would warrant a whole workbench in itself.

Visualisation is also a new concept to the Web. Much work is required in this area to help reduce readability problems and improve structuring of the Web. Currently only one tool is available. This section of the workbench merely provides a pointer to the site which developed this tool, this helps the users interested find more information on visualisation.

5.3.3 Management Section of the Workbench

This section is not very big. More research and work in conjunction with sites that host Web services would help to increase the size and thus the usefulness of the section. It does have a

useful section on guidelines and standards which are provided by the management and this was developed through research with a site hosting a Web service.

5.3.4 Future Developments of the Workbench

Future research to aid the further development of this workbench would be into development and maintenance of Java and VRML applications. Tools to aid maintenance through analysis of hyperdocuments and document generation are needed. Research into simplifying design methods for non-technical users is needed along with design methods for smaller documents which do not provide a significant overhead of work. Tools that support the implementation of guidelines would improve the quality of the workbench.

5.3.5 CCTA Trial Usage

The CCTA gave the workbench a trial use. They main problem with the workbench was that it was very simple in parts, saying that guidelines given were obvious and that in other parts it was too difficult to follow. In effect the workbench is trying to target too many audiences at one time. To have a more successful workbench would mean having one for beginners, one for technicians and one for academics. It was also felt that the workbench does not take its own advice. Otherwise it was felt that the workbench was clear and provided some useful advice covering a wide array of material.

5.4 Overall Assessment of the Workbench

The workbench achieves its goal, in that it fulfils the requirements provided by the case studies and solves some of the problems identified through the literature survey in chapter two. It does not, however, solve the problems conclusively. Much research is needed to fulfil this task. In the time available, the workbench produced a means of supporting the development and maintenance of Web documents and the management of a Web site, it did this by providing guidelines, methods and tools to the users via a Web based service.

5.5 Summary

The workbench, while satisfying its requirements had some problems. In parts where requirements specified the need for a method or tool, guidelines were supplied instead. This was the result of a lack of time allowed to satisfy each requirement comprehensively. It was not always possible to ensure that the solution that was provided for a requirement was the best solution, in many cases more research was required to decide this. The trial usage showed that the workbench did indeed provide support for development, maintenance and management,

however, much more could be done to enhance the workbench. Java and VRML seemed to be the next big areas of research for development and maintenance.

The following chapter gives an appraisal of the work as a whole, giving its strengths and weaknesses as well as describing future work to be done in this area.

CHAPTER 6

CONCLUSIONS

6.1 Objectives of the Chapter

The aim of this chapter is to conclude the work undertaken in the previous chapters. This will be done by summarising the work done and giving a critical appraisal of the work. The criteria for success will also be studied and the chapter will conclude with a section on further research to be done in the area.

6.2 Synopsis of Work

The purpose of this section is to give a brief synopsis of the work done.

This work began by looking into the area of hypertext and the World-Wide Web. In chapter two an introduction to hypertext and the WWW were given. The problems with the WWW were described. Then WWW maintenance was defined and discussed, explaining the need for WWW document maintenance. The problems that currently exist with Web maintenance are then identified. The chapter goes on to explore the previous work done in Web maintenance. The particular areas that it covers are work done with standards and guidelines, analysis of hypertext applications, including the use of metrics, tools for software maintenance and visualisation and navigation of hyperspace. The study of previous work done was supported by two case studies, one of the CCTA and one of the Durham University IT Service. Here a survey of the management structure of each organisation was done. This was followed by studying the approach adopted by each organisation to the task of maintaining and managing Web pages. An assessment of the good points and weak points of each organisations approach was given followed by some points for improvement.

The work followed on with the development of a solution. This was done by gathering the requirements of each organisation with regard to their maintenance and management needs. The requirements also took into account the development needs as it was felt that some of the maintenance problems stemmed from the development stage, e.g. poor quality of authorship.

At this point, a workbench was designed and implemented. This workbench was in the form of a set of Web pages. These Web pages contained guidelines on Web development, maintenance and management and there were some tools and methods provided with the pages for maintenance and development.

In the following chapter, chapter five, the results of a trial usage of the workbench were given. The evaluation was done by checking that the workbench conformed to its original requirements. The second part of the evaluation was done through using the workbench and discovering some strengths and weaknesses.

The work is concluded by evaluating the success of the work and identifying future research in the area.

6.3 Criteria for Success

The criteria for success were:

1. An investigation into the problems that exist in this area.

   This was mainly addressed in chapters two and three. Chapter 2 gives a literature survey of the areas of hypertext, the World-Wide Web, World-Wide Web maintenance and previous work done on World-Wide Web maintenance. After each section, problems with each area are defined. In chapter 3, there are two case studies given. These case studies identify more of the work done on Web maintenance and management. At the end of the chapter a synopsis of the problems is given.
2. An approach to overall development and maintenance of hyperdocuments that helps to solve some of the problems identified in the investigation.

   This was done through the workbench described in chapters four and five. A requirements analysis was done through the case studies and through the problems developed in chapters 2 and 3.

6.4 Appraisal

One major difficulty of the work, is that the area in question has changed a lot since the start of the work. This is because the research area in undergoing rapid change and thus some of the problems and solutions provided may quickly become outmoded. The work focuses on Web maintenance. This area is still a problem and the work done will begin to provide the start of a solution to some of the problems. There were three major areas that were needed in order to resolve maintenance problems. These were methods, tools and guidelines. The work provided for the main part, guidelines on how to carry out maintenance given the current limitations that exist. Some of these limitations will have changed, such as the existence of better development tools. However, for the most part, it appears that problems still exist. The workbench has proved to be useful in providing an approach to developers and maintainers.

6.5 Future Work on the Workbench

The workbench lacked the following items: a method for dealing with inconsistencies created through making changes, a survey of all the available design methods. It failed to produce a resulting design method that is suitable for all sizes of document and all abilities of development.

Guidelines for the maintenance of Java would be a further development. At the time of writing, not enough is known about Java to produce specific guidelines on the subject. Much more research is required into the maintenance problems and solutions associated with Java in order to produce a suitable set of guidelines.

A tool for tracking frequent problems and stopping them recurring would be a useful extension to the workbench's maintenance section. Fully automated checking of Web pages is not given in the workbench which would also be a good future enhancement. Once tools for all aspects of maintenance have been developed, an integrated tool for maintenance would be an improvement to the workbench.

6.6 Future Work in this Area

The areas that need to be worked upon fall into three main categories, tools, standards and guidelines, and methods. There are other areas that are beyond the boundaries of this work which cover the future development of hardware, HTML and the Hypertext Transfer Protocol (HTTP).

The problem with the lack of tools is ongoing. Tools that allow support for non-professional developers to manipulate hypertext properly are needed. A wider range of tools for maintenance is required. Integrated tools which perform the whole range of maintenance tasks, such as quality checking, link checking, syntax checking and document generation are needed. Tools that support collaborative authoring, solving problems encountered by distributed databases management systems, such as lost updates are needed. Tools that can check the compliance with standards and guidelines are needed. This leads on to the problem of standards and guidelines.

More standards, guidelines and conventions are required. This would be further helped by the development of metrics for hypertext.

Two areas of that will develop increasingly in the future are that of Java and VRML. The following area of work would be to develop guidelines for the maintenance of Java and VRML.

Looking at the broader spectrum, work on the hardware aspect of the Web is also needed. This includes, easy to set up and maintain servers and improved networks to allow more efficient communication between computers.

6.7 Summary

To summarise, this chapter concludes the work done in this thesis. It starts by giving a synopsis of the work. It then shows that the work fulfils the two criteria for success. This is followed by giving a description of how the workbench could be improved upon through better methods, more tools and larger sections on areas such as the maintenance of Java. The final section talks about future work to be done on Web maintenance. This involved creating standards guidelines and conventions, improved methods and better tools.

APPENDIX A

TOOL LOCATION

Tool Name	Tool Location
HTML Validators	
A Kinder Gentler HTML Validator	http://ugweb.cs.ualberta.ca/~gerald/validate
WebTechs HTML Validation Service	http://www.webtechs.com/html-val-svc
Link Validators	
html_analyzer	http://www.gatech.edu/pitkow/html_analyzer/README.html
MOMSpider	http://www.ics.uci.edu/WebSoft/MOMSpider
EIT Link Verifier Robot	http://wsk.eit.com/wsk/dist/doc/admin/webtest/verify_links.html
URL Minder	http://www.netmind.com/URL-minder/URL-minder.html
Server Log File Analysers	
Getstats program	http://www.eit.com/software/getstats/getstats.html

APPENDIX B

CCTA CASE STUDY

B. General Overview of Objectives of the CCTA with regard to the World Wide Web

This is a case study carried out in conjunction with the CCTA. This one was done by visiting their headquarters in Norwich and talking to the management team about the running the site.

B.1 CCTA Government Information Service

CGIS is the CCTA Government Information Service which has been set up to develop a World-Wide Web site for organisations within the public sector. They are there to investigate current and emerging technologies for the dissemination of public sector information. The team consists of six members and a description of the structure of the team is given below. This is followed by a detailed overview of the Web site development, a look at the future of the information service, the problems that currently exist, the foreseeable problems and the proposed work to solve these problems.

B.2 The Team Structure

The team in charge of the development and maintenance of the CCTA Web site consisted of six team members.

Team Manager

The Team Manager is the person in charge. He oversees all the other operations carried out by the other team members. He also tries to keep abreast of current developments and decides whether these should be incorporated into the CCTA Web site and if so how. He or she promotes or sells the service to organisations within the public sector.

Development Manager

The Development Manager is in charge of the development of the Web site as it grows. He or she also assists the Team Manager in researching emerging technologies and oversees the incorporation of these new technologies into the Web site. The Development Manager researches organisations to which the service can be 'sold'.

Information Manager

The Information Manager oversees putting the information onto the Web and checking that it is all correct and that it conforms, in as far as possible, with the standards laid out by the CCTA. He is assisted by the Information Manager's assistant, who puts in the information and generally helps with this side of the job. The Information Manager also visits potential and current customers and gives four hour lectures on writing Web pages giving a general overview of what it is about, how to write the Web pages and the standards expected by the CCTA. In the future the information manager is likely to be the one writing the Java applets or the VRML applications.

Operations Manager

The Operations Manager oversees the smooth running of the Web site and that all the fundamental tasks are being carried out.

Technicians (x2)

The technicians run the server. They are what is known as the server administrators. They ensure that the server is running smoothly (e.g. it is not going down regularly) and they have access to the log files which can be analysed to ensure that there are no regularly occurring faults that can be fixed by the CCTA.

B.3 The Structure of the Web Site

This aims to give an overview of the Web site and how it is run at the CCTA.

B.3.1 System put in place for the set of CCTA pages

Initially one server was set up by the CCTA, which hosts the Web site for the CCTA and all the Web pages using their service. Eventually some organisations may feel they can justify buying their own server and setting up independently.

The CCTA goes out to any areas of the public sector and offers to teach them how to create a Web document, its uses and the advantages. The organisation in question, then, either writes their own pages or gives the information to the CCTA who then write the pages for them.

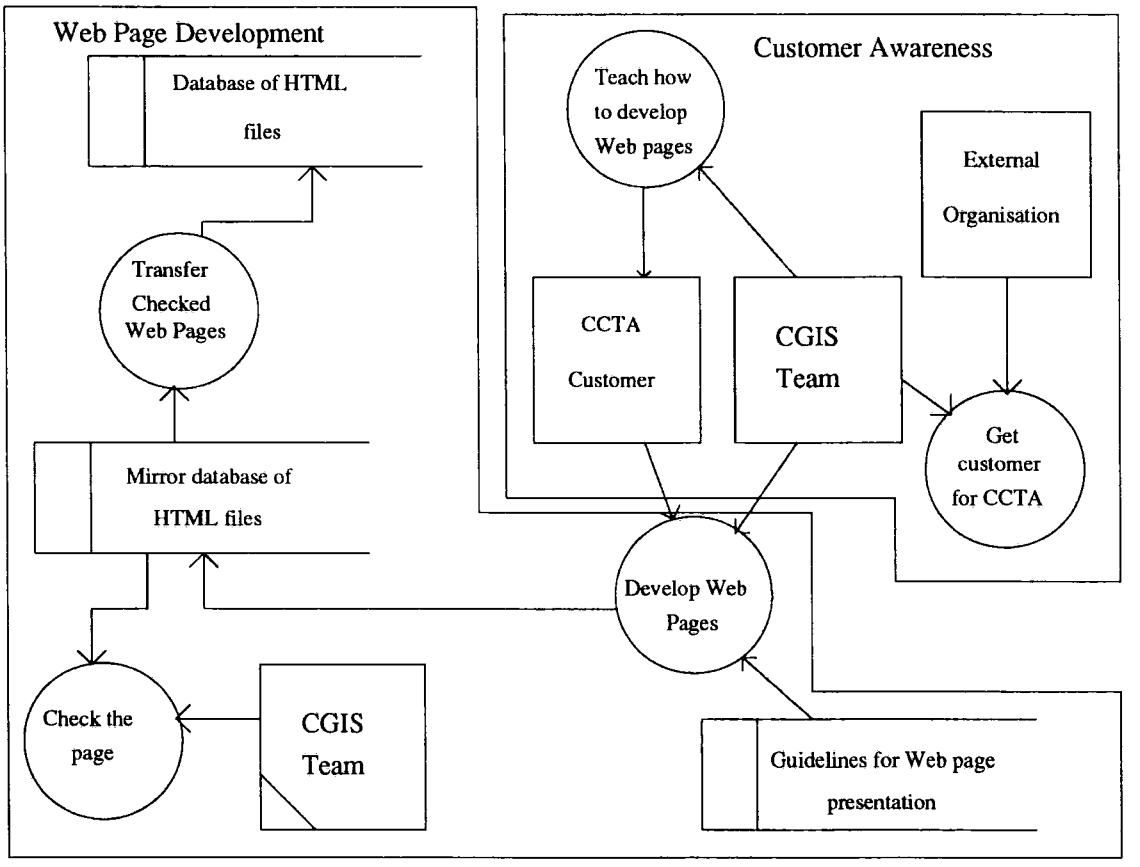


Figure 2.0 Process Model of the CGIS Team Procedures

Figure 12 CCTA management model

In the *Figure 12 CCTA management model*, a model of the way the system for managing and maintaining the site is given.

The organisation are given guidelines by the CCTA to follow. These are the following:

Style

The CCTA expect that the home page of each organisation should not contain too many graphics as they take long to download, the home page should be clear and informative and it should be obvious what the pages contain.

Structure

The CCTA expect the home page of each organisation should be simple, the pages should be centred on every browser. It should be possible to fit all the information on the home page with no scrolling required. It should be ensured that the pages look good on all screens, including black and white ones and it should be clear and straight forward to navigate your way round these pages.

Content

The pages must contain two standard CCTA buttons: one to the help and one to the CCTA home page. A 'last updated' and by whom message should be inserted at the end of the home page of each organisation. The organisation should target the audience with its information.

Once the pages are completed, they are sent in HTML or plain text to the CCTA. The pages are then checked on a mirror of the actual Web site that exists on the Novell network at the CCTA. The following checks are made:

- Are all the links OK?
- Are the CCTA buttons included?
- Is the page centred? And will it appear so on all screens?
- An overall check for glaring problems.

If all is well, the pages are then put on the server. If there are corrections to be made then the pages will not be installed until these are completed. If only small changes are required, then these will be carried out by the CCTA staff (the information manager). Large corrections are referred back to the provider organisation who is requested to make these before installation. The information manager decides on whether a correction is to be carried out by the CCTA or the organisation concerned.

Once the page is on the Web no maintenance checks are carried out.

If a change needs making to the page, or the page is updated by the organisation concerned, they send the changes or the new pages to the CCTA. The CCTA estimates a delay of two days for the changes to be made. For new pages this can be up to a week. They have a booking in system for the pages to ensure that pages that have arrived do not get lost or neglected.

The CCTA carries out no quality controls on the contents of the documents, bar their own guidelines. They view whatever is contained in the document as the concern of the organisation.

They maintain the attitude that it is best in most cases to publish the material without asking for permission from whomever concerned, and if it is a problem, they remove it. The idea being that the bureaucracy involved in continually asking permission is too great and time consuming. They have only once had to remove material, the reason being that the information was not deemed fit for public viewing as it was fairly sensitive.

B.3.2 Development of Web Site in the Future

The CCTA views the Web site as expanding continuously. As yet no 'customer' has withdrawn a Web page from the server. Customers only leave when they feel they can justify the cost and the need for their own server, as well as sufficient expertise to do so. The CCTA keep the links in the 'Index By Organisation' page to preserve the structure and completeness on the CCTA Web site. So as more organisations request to have Web pages and more organisations become independent with their own servers, they preserve the number of sites and the work load for the CCTA on a reasonable level while the site grows. Currently they have 120 customers using their service. There are now fifteen government servers in use and 4 customers have left and set up there their own sites, they have, however, requested to remain linked to the CCTA pages.

The future will inevitably lead to the usage of Java applets and VRML. The CCTA intend to try and persuade customers to start using Java applets as part of their Web pages. Currently the CCTA staff are learning to program Java and they will initially write the applets for the customers. This will be presumably be followed by lectures to the customers from the CGIS team on how to program Java applets.

3.2.3.6 The Tool

The CCTA have produced a tool for the maintenance of their Web site. The tool is known as the Web Manager. It uses the information contained in the Web to build an information database of all files and links. It then tracks down broken links and reports them. Web Manager is a Microsoft Windows based link verifier tool. The choice of the Windows environment for the tool the CCTA claims is due to the growing number of development tools which are Windows based. Web Manager is based on a relational database and maintains a record of all entities(Files, HREFS, NAMES, URLs etc.) in a Web. The database is built automatically from an analysis of all files in the Web. Web Manager supports multiple Webs.

If you have Webs for many different customers, Web Manager provides information by customer. Web Manager provides multiple information windows, for example:

- HTML files sorted by directory, file name, date and title;
- GIFs files including version and interlace information sorted by directory, file name, date and size;
- URLs, FTP and MailTo references
- Broken file links
- Broken graphics links

Web Manager maintains a database of broken links both internal, within a file and external, between files. Web Manager maintains a database of unused files. Old files which are no longer referenced and can be safely deleted or, more importantly, files which should be part of the Web but have no hypertext links pointing to them. Web Manager is not an HTML analysis tool. It makes no judgements about the validity of the HTML it reads. Some badly formed links, which are acceptable to a Web Browser, can result in Web Manager reporting the link as open. Most browsers are not as strict as Web Manager.

Web Manager supports a wide range of printed reports, for example:

- HTML File Report
- Graphics File Report
- Off Site Links Report
- Open Link Report
- Unused File Reports
- Complete File Map of all Web Links.

Future Enhancements of the Web Manager

A future version of Web Manager will detect mixed case links. This type of link is typically found in HTML documents written in a DOS/Windows environment. These links work successfully when tested in this environment but break when transferred to a live Web site on a UNIX machine. A future version of Web Manager will detect absolute file links. This type of link works in the authoring environment in its defined directory structure but fails when moved to a live Web site with a different directory structure. Web Manager does not currently support the validation of off site links (URL, FTP, MAILTO etc.). This feature will be added to a future version. It intends to take into account Java applets and VRML though quite exactly what functions it will perform were unclear. Web Manager requires Windows 3.1, Windows for Workgroups or Windows 95. Web Manager has not been tested on Windows NT. Web

Manager requires 6 Mbytes of disc space for program files and additional database space depending on the size of Web.

Unfortunately, I was unable to see the Web Manager in use during the visit as it was not fully working. It is for this reason that I cannot provide an evaluation of the tool and the above is merely a brief description provided by the CCTA.

APPENDIX C

IT SERVICE CASE STUDY

C.1 Information Technology Service

The IT Service are there to ensure that services which require the use of information systems are properly decimated and supported throughout the university. In this respect they have a similar role to the CCTA, who also provide an 'IT service'.

The WWW service is said to be primarily as a support to academic activities. This objective should influence decisions made about resources, information on the Web and so forth.

C.2 The Personnel

Unlike the CCTA, the IT Service does not have a team to support the running of the Durham University Web site. Instead one person is responsible for the Durham University Web site. We shall call this person the site manager. The site manager has two jobs, one to manage the Durham University site and the other managing the IT service Web site which provides more information about using the Web. The site manager will also liase with corporate communications committees, in order to assess such things as the corporate image presented by the Web pages.

C.3 The Structure of the Web Site

C.3.1 System put in place for the set of Durham Web pages

The University Web site is a more complicated and has a larger infostructure than the CCTA sites. Each user, (person having access to the university UNIX computing facilities), may have a personal Web page. This is subject to rules and guidelines but is not perceived to be part of the university Web site, unless for example a department links a personal page into their set of pages. This will be elaborated upon later. The university Web site consists of a main university

page and IT Service set of pages. Departments can have, but not all do, departmental pages, colleges and societies also have pages. These are all (in as far as possible) linked to the main university pages. So the current procedure is to apply for a username for yourself, department, society or college, then if you are a society, department or college you can inform the site manager of the pages' existence and these will be linked into the main university pages. Official pages: academic departments, inter-departmental research groups, service departments, Colleges, DSU-affiliated student societies, college JCR's will explicitly be linked to the university home page.

All Web pages that are linked to the Durham Web pages via the ITS site manager. All of the university have access to a live Web, if they have a UNIX username and password, however, linking to the Durham university Web pages needs to be done via the site manager. There are rules for official and personal pages. This information is provided on the Web. Enforcing rules is the job of corporate communications. Soon there will be an internal and external Web. The ITS is currently going through a restructuring of the system. There is no official use of tools and maintenance consists only of checking pages when they are added. Periodic maintenance is not carried out.

Regulations

In order to preserve a style or university feel to the pages which will be linked to the main page and to be able to distinguish between personal pages and university pages a set of guidelines and a code of practice are put in place.

All pages made using university facilities must either be an official university page or a personal page, no other is allowed.

Failure to observe any of the following points is construed as misuse of university facilities and appropriate action is taken by the disciplinary board.

- Offensive, inappropriate or illegal material must not be included.
- Potentially defamatory material which could be held to impugn the reputation of people or organisations must not be included.
- On personal pages, personal opinions should not be expressed as if they were the opinion, or policy, of the University, or any other organisation.
- Personal pages must not use the University name or crest as a header or title.
- Official pages must conform to a set of rules of presentation provided by the University.

Rules of Presentation of Official University Pages

- Official University pages must name the University, the department or group to which the page pertains, and a contact name for the person responsible for the page.
- These pages should display the University crest as provided by files held in local icon libraries
- Official pages should include a standard set of navigation aids, providing easy access to the University Home Page and the relevant departmental page.
- Official pages should conform as much as possible to the recommended guidelines on presentation.

Guidelines for Presentation

There are on-line style documents for further reading provided.

Graphics, images, colour and multimedia:

- Remember not all browsers display images,
- remember that images increase network traffic considerably and consequently the time to access your page,
- remember that some browsers may require external software for a particular media and this software may not be available.

Page and document size:

Browsers determine the final appearance of a page - this affects which fonts are use and spaces between lines, They will also determine the size of the page. So do not think of the page as a fixed size.

Document Size

- Make use of the HTML built in headings, but don't add your own white space, this can be ignored by some browsers.
- Use HTML links for cross-referencing and avoid explicit section numbers where possible.
- If there is a natural hierarchy to the information, use this to aid navigation through the document.
- Use links between the documents where relevant and include pointers to related documents.
- Remember that you are imposing your own perceived structure and the reader may not follow the same path as you expect.

Text Layout

- You do not have control over the final layout of your pages. You can use specific bulletting from HTML.

Content

It is easy to produce material for publication on the Web which will reach a much wider readership that paper-based material and other on-line material.

- Make sure the content of all pages is reviewed and edited to a high standard.
- Delay publication until the accuracy is assured.

Corporate Image:

- Official university pages should include the University crest as provided in the local icon library.
- Personal pages should not display the University crest.

Character sets and accents

HTML has built in character entities, but browsers vary in their ability to represent these.

- Where possible it is probably best to use the standard ASCII character set, but this situation will change as the HTML standards develop.

Authorship/Editorship

- Always provide a contact name for feedback, notification of errors and praise, either using the form facilities available on some servers, or by giving an e-mail address of the author or editor of the information. *editor is the maintainer...*
- Provide some idea of the currency of the information - at least by providing a date at which it was last updated.
- Provide and indication of the status of the page (e.g. draft, consultative, confirmed..) and any copyright restrictions on the content.
- Place this information in a consistent location in all your pages.

Navigation tools

- Provide navigation tools wherever appropriate, (e.g. to the University home page, to the next page, to contents, to index and son on).
- Icons can be used, but be consistent between pages.

- Be careful about using phrases such as 'click here'.

C.4 Development of the Web Site in the Future

At the time of the case study, the Web pages were all one unit representing the university, however, the plan for future was to have two Web sites. One would be an internal Web site, that was for the use of members of Durham University and the other would be representative of Durham University and be an external site. The external site would be more of a prospectus for Durham University, geared towards advertising the place. This would solve some of the problems of targeting the right audience. It would take away the problem of presenting the wrong image to the wrong people. It will, however, almost certainly require more people to manage the two sites. Another development of the future, may be that when the societies and departments for the most part have a Web site, the management may well seek to encourage the remaining few to set up Web sites.

C.5 Tools

The do not make use of any tools for maintenance, however they do make tools available for use of the users. These tools are development tools and browsers, The following development tools are available: HoTMetaL, HTML Assistant and the Microsoft Word extension which is also a browser. The browsers available are Netscape, Mosaic, Lynx.

ACKNOWLEDGEMENTS

I would like to thank the following people:

My parents, for being very patient with my interminable education and providing the necessary financial backup. It's over now I promise!

Clive, for providing practical help such as getting it bound and printed and moral support when it was most definitely needed, but most of all for always believing in me when I wasn't all that convinced. Oh and all the red wine...hic!

B & R, also for moral support and entertainment! By the way, nice window!!!

My supervisor, Cornelia Boldyreff, for the help, support and useful advice.

The CCTA and the Durham University IT Service for their co-operation with the case studies.

I would like to thank Liz Burd for her support with coping with the obstacles that cropped up regularly and for providing invaluable support and advice at getting over these hurdles, but most of all for keeping me up to date with goings on and the gossip!!

I would like to thank Pui-Shan for putting up with me all year and being an entertaining office mate.

I would like to thank Richard and Nigel for their help with my numerous technical problems.

REFERENCES

[1] Lincoln D. S., *How to set up and maintain a World Wide Web Site: The Guide for Information Providers*, Addison-Wesley, 1995.

[2] Conklin, J. *Hypertext: Survey and Introduction*, IEEE Computers, pages 17-41, 1987.

[3] Bieber M., Isakowitz T., *Designing for Hypermedia Applications*, Communications of the ACM, Vol. 38, No. 8, August, 1995.

[4] Berners-Lee T., Cailliau R., *World-Wide Web: Proposal for Hypertext Project*, CERN Memo, http://www.w3.org/hypertext/WWW/Proposal.html, November, 1990.

[5] Berners-Lee T., Cailliau R., *World-Wide Web*, invited paper: Computing in High Energy Physics, annecy, France, 23-27 September, 1992.

[6] Berners-Lee T., Cailliau R., Luotonen A., Nielsen H. F., *World-Wide Web*, Communications of the ACM, pages 76-82, Vol 37, No 8, August 94.

[7] Berners-Lee T., Cailliau R., Groff J. F., B. Pollermann, *World-Wide Web: An Information Infrastructure for High Energy Physics*, Software Engineering, Aritificial Intelligence and Expert Systems for High-Energy Physics, January, 1992.

[8] Relihan L., Cahill T., Hinchey M. G., *Untangling the World-Wide Web*, http://itdsrv1.ul.ie/research/WWW/utwww.html, 1994

[9] Fielding R. T., *Maintaining Distributed Infostructures: Welcome to MOMspider's Web*, First International World Wide Web Conference, May, 1994.

[11] Flohr U., *Hyper-G Organises the Web*, BYTE magazine, November 1995.

[12] Hatzimanikatis A.E., Tsallidis C.T, D. Christodoulakis *Measuring the Readability and Maintainability of Hyperdocuments* Software Maintenance: Research and Practice, Vol 7, 77-90 (1995)

[13] Brown P., *Assessing the Quality of Hypertext Documents*, Proceedings of the European Conference on Hypertext, France, November 1990

[14] IEEE *Standard Glossary of Software Engineering Terminology*, IEEE, 1983.

[15] Bennett, K. H., Robson, D. J., Munro, M., *Software Maintenance* in Software Engineering Reference Book, McDermid, J., Butterworth Heinemann, 1991.

[16] Isakowitz T., Stohr E. A., Balasubramanian P., *RMM: A Methodolgy for Structured Hypermedia Design*, Communications of the ACM, Vol 38, No8, pp34-44, August 1995.

[17] Kaplan, S. M., Maarek, Y. S., *Incremental Maintenance of Semantic Links in Dynamically changing Hypertext Systems*, Interacting with Computers, Butterworth and Heineman, 1990.

[18] Halasz F., *Reflections on notecards: Seven Issues for the Next Generation of Hypermedia Systems*, Communications of the ACM, Vol. 31, July 1988.

[19] Garzotto F., Mainetti L., Paolini P., *Hypermedia Design Analysis, and Evaluation Issues*, Communications of the ACM, Vol. 38, No 8, August 1995.

[20] Gershon N., *Moving Happily through the World Wide Web*, Computer Graphics and Visualization in the Global Information Infrastructure, CG&A, Vol 16, No 2, March 1996.

[21] Kahn P., *Visual Cues for Local and Global Coherence*, CACM, Vol 38, No 8, August 1995.

[22] Wood A., Drew N., *Hyperspace*: http://www.cs.bham.ac.uk/~amw/hyperspace, 1995.