Application of Machine Learning Algorithms in Adaptive Web-based Information Systems

Adalet Serengül Güven Smith

School of Computing Science
Middlesex University
United Kingdom

Thesis submitted in partial fulfilment of the requirements of the degree of Doctor of Philosophy in Computer Science

November 1999

Abstract

Hypertext users often face the difficulty of identifying pages of information most relevant to their current goals or interests, and are forced to wade through irrelevant pages, even though they know precisely what they are looking for.

In order to address this issue, this research has investigated the technical feasibility and also the utility of applying machine learning algorithms to generate personalised adaptation on the basis of browsing history in hypertext. A Web-based information system called MLTutor has been developed to determine the viability of this approach. The MLTutor has been implemented, tested, and evaluated. The design of MLTutor aims to remove the need for pre-defined user profiles and replace them with a dynamic user profile building scheme in order to provide individual adaptation. This is achieved by a combination of conceptual clustering and inductive machine learning algorithms. This integration of two machine learning algorithms is a novel approach in the field of machine learning.

In the initial prototype of MLTutor, a simple attribute based conceptual clustering algorithm and the ID3 algorithm were implemented. An assessment of the initial prototype highlighted the need for an in-depth investigation into the machine learning component of the prototype. This investigation led to the development of a multiple decision learning algorithm named SG-1 and a scheme for attribute encoding within the system.

In order to assess these enhancements a comparative study was conducted with four adaptive variants of MLTutor along with the non-adaptive control. The adaptive variants were developed to allow alternative approaches within the machine learning component of the system to be compared. Two of the variants applied the clustering algorithm dynamically and used two different cluster selection strategies. These strategies were based on the last page visited and a weighting of recently visited pages. The other adaptive variants used pre-clustered data with the same cluster selection strategies. The comparative evaluation undertaken on the variants used a number of established evaluation criteria and also introduced an original cross analysis scheme to determine how the adaptive component of MLTutor was utilised to complete a set of tasks. This cross analysis scheme highlights a number of weaknesses related to the evaluation methods commonly used in the field of adaptive hypermedia. The results have also highlighted a technical limitation with the particular clustering algorithm employed, specifically the generation of a heterogeneous cluster that results in poor suggestions in some circumstances.

The results of the evaluation show that the MLTutor is a functional and robust system. Although the utility of using machine learning algorithms to analyse browsing activity in a hypertext system is unproven, the technical feasibility has been established.

Declaration

The research presented in this thesis is entirely my own. In the course of carrying out this thesis research, the following publications were produced.

Guven S. (1999). 'MLTutor: A Web-based educational adaptive hypertext system' *Proceedings of IEE Colloquium on Lost in the Web - Navigation on the Internet*, November 1999, IEE Savoy Place, London.

Guven S. and Blandford A. (1998). 'MLTutor: A Symbolic Multi-Algorithmic Approach to Leaner Profiling in WWW-Based Educational Hypertext System' *European Conference on Machine Learning, ECML'98 Workshop: Learning in Humans and Machines,* April 1998, Chemnitz.

Guven S. and Blandford A. (1998). 'An adaptive WWW-based Educational Hypertext System using Symbolic Machine Learning' *Proceedings of Third Middlesex University Conference on Research in Technology*, April 1998, ISSN 1462-0871, Middlesex University.

Acknowledgement

There is a strong temptation to try to individually acknowledge everyone who has helped me to reach this point. Sadly, such a list would be much too long and it is inevitable that many people would be unintentionally omitted. I would thus like to thank everyone who, knowingly or otherwise, has provided support, encouragement and assistance along the way. However, there are others who have played a major role in the production of this thesis and who deserve a more personal note of gratitude.

Firstly, I wish to express my deepest gratitude to my supervisors Prof. Ann Blandford and Prof. Steve Torrance for their excellent guidance, concern and endless support during the course of my research study.

I also wish to thank Dr. Ian Williams, Dr. Huw Jones and Lian Scholes for their tremendous help during preparation for the empirical work and Dr. Kevin Boone, Dr. David Westley and Paul Gillary for their advice on statistical methods. I would also like to thank Dr. Paul Cairns, Dr. Chris Hayck, Prof. Harold Thimbleby, Prof. Ian Witten and my external advisor Prof. Alan Hutchinson for their helpful comments, advice and discussions. Special thanks also go to fellow students who took part in empirical study.

Many thanks also to Dr. Yin Leng Theng, Dr. Cécile Rigny, Skip Basiel, John Okyere-Boateng, Thomas Tan, Doreen Ng, Kok Fong Tan, Günay Akin and Dursun Yildirim for their support and great friendship. My very special thanks go to Penny Duquenoy and Jo Hyde for being a constant source of encouragement and support throughout.

To Michael Smith, I owe thanks beyond measure for his tremendous help and patience.

Most of all, my deepest thanks go to my parents for their unconditional love and support.

Table of Contents

Chapter 5: Machine Learning

Chapter 6: MLTutor Overview

Chapter 7: Initial Implementation and Investigation

Chapter 8: Evaluation Framework

Chapter 9: Results

Chapter 10: Discussion and Conclusion

Table of figures

Table of tables

Chapter 1

Introduction

1.1 Introduction

The idea of an intelligent personal assistant is not new. In the early 80's Michalski (1980) anticipated that individuals, in the expanding information society predicted by Bush (1945), the father of hypertext, would need intelligent personal assistants to cope with the overwhelming amounts of available information and the complexity of every day decision making. In addition, as stated by Michalski (1980), the knowledge and the function of such (computer based) assistants should be dynamic in order to adapt themselves to changing demands; in other words any such systems should be able to learn.

Traditionally, most research on computer based (machine) learning has dealt with the development of techniques for solving engineering problems and many of the systems developed have been tested on simplified artificial problems (Reich 1994). Consequently, the machine learning research field has historically been very rich in terms of theoretical developments but lacks practical applications with direct links between theory and practice. This is particularly noticeable in the field of adaptive educational hypertext.

The research presented in this thesis is an attempt to bridge the gap between theory and practice in the domain of WWW-based systems. The World Wide Web is an excellent mechanism for the dissemination of information and a few WWW-based systems can provide support by adapting material to the needs of a user. A number of these systems employ machine learning techniques (NEWS WEEDER (Lang 1995); IAN (Green 1995); Magi (Payne and Edwards 1997); WebWatcher (Armstrong *et al* 1995); LAW (Edwards *et al* 1996)). The key challenge in such systems is to be able to capture an individual user's preferences and specific information needs and utilise this information to adapt the environment to the user.

However, in many recent systems the general principle behind the form of adaptation employed has largely been based on mapping users' specifications onto pre-defined system user profiles (Syskill and Webert (Pazzani *et al* 1997)). The development of such profiles is a very time consuming and laborious task and most prototypes are restricted to one domain (Edwards *et al* 1997) and the knowledge bases of these systems are likely to be "hand-crafted" (Hohl *et al* 1996).

In this thesis, as an alternative, a domain independent machine learning component for building dynamic user profiles has been implemented and its utility investigated. This approach facilitates a flexible, individualised approach to adaptation without the need for additional input from a user or pre-classification of users.

1.2 Motivation

The starting point for this research is my interest in the application of machine learning algorithms. This particular interest began during my MSc dissertation (Smith *et al* 1996) at Middlesex University between September 1993 and February 1995. During my work placement, undertaken as part of my MSc at the National Institute for Medical Research (NIMR) laboratory in Mill Hill London, I had an opportunity to investigate the viability of using machine learning techniques for parameter selection in multiple protein sequence alignment in the domain of molecular biology. The positive results of this work encouraged me to seek out further domains for the application of machine learning.

I was offered an opportunity to pursue this goal by the School of Computing Science at Middlesex University when they accepted my application to study a PhD. The School of Computing Science at Middlesex University has an active research programme in the fields of Hypertext, Multimedia, World Wide Web (WWW) developments, Human Computer Interaction and Intelligent Educational Systems.

Initially, I was intrigued by the possibility of integrating machine learning techniques into an intelligent educational system with a view to aiding student model diagnosis (Self (1987); Costa *et al* (1988); Gilmore and Self (1988); Elsom-Cook (1988); Chan and Baskin (1988); Self (1990); Woolf and Murray (1991)). My research scope was subsequently broadened by the work of a colleague's research into the problems associated with hypertext (Theng 1997).

Aware of some of the problems users have finding appropriate information in hypertext and WWW-based distance learning, I recognised that machine learning techniques offered a way of improving user interaction within WWW-based educational systems.

1.3 Objectives

The objective of the research was to design, implement and test a prototype system in order to assess the feasibility and the utility of using machine learning algorithms to analyse a user's browsing patterns without the need for pre-defined user profiles, with a view to dynamically providing adaptive navigational support in a WWW-based educational system.

The prototype system developed has been named MLTutor. The machine learning algorithms used in the initial prototype of MLTutor were a simple conceptual clustering algorithm

(Hutchinson 1994) and the inductive learning algorithm ID3 (Quinlan 1986). Both of these algorithms process attribute based descriptions of objects: Hypertext pages in this case. The role of the conceptual clustering algorithm was to find inherent patterns based on the attribute encoding within the hypertext pages visited by a user and so eliminate the need for pre-defined user profiles. The ID3 algorithm was employed to interpret the information concealed in clusters in the form of attribute based rules. These rules are used to provide adaptive navigational support in the form of a list suggesting hypertext pages the user may be interested in.

This initial combination of algorithms was tested in a small-scale experiment using a 32-page educational document, with six participants and various attribute configurations. This experiment highlighted a number of weaknesses, which led, through a process of refinement and further investigation, to the development of the SG-1 algorithm, an enhancement of ID3, to replace ID3 in MLTutor. This further investigation also led to the formation of principles for the attribute encoding of hypertext pages.

In order to evaluate the enhanced MLTutor, a more significant empirical study was conducted. For this study, four adaptive variants of MLTutor along with a non-adaptive control were constructed. The adaptive variants incorporated alternative cluster selection strategies based on last page visited and the page weighting mechanism. Two of the variants incorporated a pre-clustering strategy without SG-1 and the other two a dynamic clustering strategy with SG-1. These variants along with the non-adaptive version were used in a comparative study in which thirty participants took part.

The goal of this evaluation was firstly to identify if the adaptive MLTutor provided benefits over the non-adaptive control, and secondly, to determine if any one of the adaptive variants was superior. The results of the evaluation were suggestive, but inconclusive. The findings suggest that the adaptive MLTutor produced better results in terms of the evaluation criteria used, however, the findings are inconclusive due, at least in part, to the small number of participants who took part in the evaluation.

1.4 Structure of Thesis

This thesis is divided into four parts: a literature survey, the design and implementation of MLTutor, the conclusion and supplementary material.

The purpose of the literature survey, presented in chapters 2 to 5, is to provide details of

published research relevant to this study.

Chapter 2 briefly presents the developments in intelligent educational systems covering electronic textbooks, Intelligent Tutoring Systems (ITSs) and more recent WWW-based educational hypertext systems.

Chapter 3 presents the key developments in hypertext research and outlines the advantages and the problems associated with hypertext, along with a number of solutions. In this chapter, the potential benefits of hypertext technology in an educational context are also discussed.

Chapter 4 reviews the methods and techniques developed in the field of adaptative hypertext research to address the hypertext problems outlined in Chapter 3.

In Chapter 5, detailed technical material on machine learning algorithms and the use of these techniques in early intelligent educational systems, and more current WWW-based educational systems, are presented.

In the second part of the thesis, presented in chapters 6 to 9, the development of a machine learning based educational hypertext system is described.

Chapter 6 provides a high level overview of the design principles underlying the MLTutor research prototype, which has been developed during this study. This chapter draws a theoretical design and implementation framework describing the use of a novel combination of machine learning algorithms to provide navigational adaptation within a Web-based hypertext system.

In Chapter 7, the results of a preliminary evaluation are presented. Based on the findings of this study the machine learning component of MLTutor was revisited which led to the development of the SG-1 algorithm. The SG-1 algorithm, along with the implementation of four variants of the machine learning component of MLTutor, are also described in this chapter.

Chapter 8 details the evaluation framework developed to measure the effectiveness of the adaptation provided by the machine learning component of MLTutor. This framework was developed following a review of the strategies employed by other researchers in the field of adaptive hypertext systems which are also outlined in this chapter.

Based on the framework described in Chapter 8, Chapter 9 details the analysis of the results obtained from the empirical study conducted using the variants of MLTutor described in Chapter 7.

The conclusion to the thesis, in Chapter 10, draws together the threads from previous chapters aiming to present a critique of this research with respect to the contributing domains.

Chapter 2

Intelligent Educational Systems

2.1 Introduction

The MLTutor (see chapter 6) is an adaptive hypertext-based information system designed for use in an educational context on the World Wide Web (WWW). The MLTutor uses machine learning algorithms (see chapter 5) to analyse a user's interaction with a hypertext to search for patterns within the accessed material. Any identified patterns are used to generate a list of related topics, which the user is free to select from a selection list. Entries in the selection list can be treated as though they are additional hypertext links available for next page selection.

The MLTutor prototype is not a tutoring system. However, it has been designed within an educational context to support task-oriented learning activity. The essential goal of MLTutor is to enhance the educational experience of hypertext users by providing details relevant to their current browsing session.

This chapter briefly presents an overview of educational systems research and indicates the role of a student / user model in adaptive systems.

2.2 Educational systems

The textbook has long been the staple of classroom education, either within the classroom itself or privately at home. Initial attempts to provide computer-based education consisted of implementing *electronic textbooks* - essentially presenting existing linear text in electronic form with minimal additional benefits to the student. Rada and Murphy (1992) claimed that experiments with electronic encyclopaedias such as Emacs-Info, Guide (Brown 1987), HyperTies (Shneiderman and Kearsley 1989) and a variant of SuperBook (Littman 1991) called MaxiBook, showed that paper based representations were preferred by some users; however certain search tasks were performed better with the electronic systems.

Although there have been significant developments in intelligent tutoring systems research (Burton and Brown (1982); Clancey (1983); Kobsa and Wahlster (1988); Ritter (1997)), most of the systems developed have been used by a limited number of students and remained as research prototypes (Brusilovsky 1995). This is partly due to the fact that most intelligent tutoring systems have been platform dependent requiring quite powerful computers to operate them.

The WWW has the potential to overcome these limitations and provide additional benefits. The WWW is widely accessible, requiring minimal hardware, and standards for access already exist.

A Web-based system could be accessible to a wide, possibly global, audience allowing for distance learning, but could equally be used within a classroom.

However, there have been limitations associated with the technology of the WWW. The WWW has traditionally been based on a simple and static hypertext paradigm allowing pages of information to be retrieved from a server, over the Internet, to a client web browser with limited facilities for interaction. The situation has changed with the advent of sophisticated scripting languages and the platform independent programming language JAVA which allows executable programs to be downloaded with hypertext pages. These programs run within the client browser and can communicate with the server. Execution of sophisticated programs on the server can be requested by the client with data passed to, and received from, the server.

These facilities have allowed educational software developers to port existing systems to the WWW. Although modifications are usually necessary to take account of the distributed multiple user environment of the web, a significant investment in existing intelligent tutoring systems can be preserved. Examples of established intelligent tutoring systems ported to the WWW include ELM-ART (Brusilovsky *et al* 1996a), a web-based version of the LISP tutor ELM-PE. (Weber and Mollenberg 1995) and PAT OnLine (Ritter 1997), a Web ported version of the PAT Algebra Tutor (Koedinger *et al* 1995).

With the rapid development of the WWW and take up of Internet technology within organisations in the form of Intranet systems, the number of Web-based tutoring systems is likely to increase. The possibility of having a number of tutoring systems focusing on the same, or closely related topics, is envisaged, and researchers (Brusilovsky *et al* 1996b; Ritter 1997) have explored ways of integrating discrete educational systems on the WWW.

This work has led to the development of a system integrating the PAT OnLine tutor (Ritter 1997) with InterBook (Brusilovsky and Eklund 1999). Within this collaboration InterBook is used to deliver conceptual information while PAT OnLine provides facilities for interactive problem solving. A key issue for collaborating systems is information sharing. Brusilovsky *et al* (1997) suggest a centralised architecture is most appropriate for such systems and they employ a centralised user modelling server within their PAT OnLine / InterBook collaboration. All collaborating components of the system are expected to access the server for the current picture of the student and update the server with details of any relevant changes.

2.3 Student modelling

The student model is a key component of an adaptive system and has featured significantly in intelligent tutoring system and adaptive hypertext system research where the term *user model* is preferred. McCalla and Greer (1991) suggest that in order to make any learning environment adaptable to individual students, it is essential to implement a student model within the system. The model should permit the system to store relevant knowledge about the student and use this stored information as the basis for adaptation of the learning material to the needs of students.

The MLTutor system, described in Chapter 6, uses a historical trace of recent browsing activity in a hypertext system to dynamically generate rules summarising the interaction. Machine learning algorithms are employed to generate the rules which are subsequently used to suggest related pages.

2.4 Conclusion

This chapter has briefly presented developments in intelligent educational systems covering electronic textbooks, Intelligent Tutoring Systems (ITSs) and more recent WWW-based educational systems.

The student model is a fundamental component of an adaptive system and is used in both intelligent tutoring and adaptive hypertext systems. Adaptive hypertext systems are discussed in Chapter 4 following an introduction to hypertext concepts in Chapter 3.

The MLTutor uses machine learning algorithms to search for patterns within the content of accessed WWW pages. The aim of MLTutor is to enhance the educational experience of users by providing details of additional pages relevant to their current browsing session. Although not an intelligent tutoring system, the dynamically generated suggestion rules created by the machine learning component of MLTutor could form the basis, or be used to update, a student model. Consequently the MLTutor could form an add-on collaborating component within an educational system which contains a hypertext element.

Chapter 3

Hypertext

3.1 Introduction

This chapter outlines the early developments in hypertext research. The many benefits and problems associated with hypertext, along with solutions to the problems, are discussed. Hypertext concepts are presented here as educational material in the form of HTML formatted documents from the World Wide Web (WWW) form a core component of the MLTutor prototype.

MLTutor is an adaptive hypertext system, which aims to assist users browsing hypertext documents for information. The MLTutor system is fully discussed in Chapters 6 and 7. Adaptive hypertext, which evolved from basic hypertext, is discussed in Chapter 4.

3.2 Definition of hypertext

In contrast to a paper representation, a *hypertext* representation provides a non-sequential method of representing and accessing information. In a hypertext document information is stored as a network of nodes connected by hypertext links. The selection of a hypertext link allows a jump to another part of the document or even to another document. A hypermedia[1] system is an extension of this principle that integrates elements of multimedia, allowing selection of animation, video and sound from within the document.

In the 1960's the pioneer of hypertext, Ted Nelson, defined the terms *hypertext* and *hypermedia*.

Hypertext: *"I mean, non-sequential writing – text that branches and allows choices to the reader,...this is a series of text chunks connected by links which offer the reader different pathways..." (Nelson 1987).*

Hypermedia: *"Hypermedia simply extends the notion of the text in hypertext by including visual information, sound animation and other forms of data..." (Nelson 1987).*

The ideas underlying hypertext have been around for quite a while; for instance dictionaries and encyclopaedias can be viewed as consisting of information chunks connected by cross-referential links. A hypertext system consists of chunks of information electronically stored in a network of interconnected nodes. As computers have become commonplace the amount of

[1] This thesis is principally concerned with hypertext and the term hypertext will be used in preference to hypermedia; however, many of the ideas expressed are equally applicable to hypermedia systems.

information stored electronically has increased massively and with the advent of popular graphical user interfaces, interest in hypertext and hypermedia has exploded. The popularity of hypertext as a means of disseminating information stems from the fact that such systems are easy to use and allow rapid and unrestricted access to information. This is particularly evident in the popularity of the World Wide Web (WWW).

The WWW, also known simply as the Web, consists of millions of hypertext pages residing on thousands of Web server computers across the Internet. From the commercial point of view, compared to more traditional means of information dissemination, Web pages are relatively easy to create, publish and update. From an educational point of view, the WWW offers great potential for learning and teaching (Laurillard *et al* 1998; Dwyer *et al* 1995).

3.3 Key stages in the development of hypertext

The roots of hypertext lie in the work of Vannevar Bush (1945) who was concerned about the explosion of scientific literature which was making it impossible for even specialists to follow new developments in their field. He envisaged that in the near future there would be a need for a new mechanism that would help people to store and access information more easily.

Based on this idea Bush sketched the outlines of a device which he called MEMEX (Memory Extender). The proposed system was a device in which an individual would be able to store all books, records, and communications on microfilm, and which was mechanised so that it might be consulted with speed and flexibility. The MEMEX was envisaged as an enlarged intimate supplement to memory (Bush 1945).

Although never implemented, it was envisaged that a MEMEX would store information on microfilms which people would be able to access using projectors from their desks. An important feature of the MEMEX device was *associative indexing*. Associative indexing allows items to be associated such that an item may cause another item to be selected automatically.

In the 1960s, Engelbart and Nelson elaborated on Bush's concept of the MEMEX. Ted Nelson (1965) claimed that "Systems of paper have grave limitations for either organising or presenting ideas. A book is never perfectly suited to the reader; one reader is bored, another confused by the same pages. No system of paper - book or programmed text - can adapt very far to the interests or needs of a particular reader or student". Engelbart and Nelson envisaged computers as building and manipulating interconnected bodies of text. In 1965 Ted Nelson, for the first

time, coined the word 'hypertext'.

Despite the early work by Bush and Nelson it was not until the mid-1980s that computers became powerful enough to support substantial hypertext features. A system of this era was the Symbolic Document Examiner (Walker 1987). The primary function of this system was to allow instant access to the contents of reference manuals for the Symbolics workstation via a hypertext interface.

Although not a success because it was not an open system (Baecker *et al* 1995), the Xerox PARC Notecards system (Halasz 1988) allowed blocks of text and graphics to be linked together. Within this system index card sized blocks of text and graphics, called Notecards, are connected by typed links as a semantic network. Notecards is a tool for displaying, modifying, manipulating and navigating through the network.

In 1987 HyperCard was released and provided free with Apple Macintosh computers. HyperCard consists of an interface builder and a scripting language that allows hypertext systems to be quickly built. The success of HyperCard led to a wider familiarity and greater acceptance of the hypertext concept.

In 1992 Tim Berners-Lee (Berners-Lee 1992) developed a hypertext system for the thousands of physicists at CERN in Switzerland to make it easier for them to write and share reports and scientific papers. The system was so useful that he made both his server and his text browser freely available to anyone who wanted a copy. The server responded to document requests by delivering a file to the browser. The server used the HTTP protocol that was very similar to the file transfer protocol (FTP) and used ASCII commands. Documents were written in the HyperText Mark-up Language (HTML). The widespread uptake of the software led to the foundation of the World Wide Web (WWW).

3.4 Building blocks of hypertext - links and nodes

A hypertext document consists of interlinked units of information. Each unit of information is stored in a *node*, and a node may contain pointers to other units of information. These pointers are called *hypertext links* or *hyperlinks*. A hypertext link typically connects two nodes, in other words it points from an *anchor* point in a *source* node to another node called the *destinotion* node (Nielsen 1995b). A particular anchor typically points to a single destination; however, many anchors may point to the same destination.

Although a hyperlink is anchored at a particular location in the source node its destination can be another location within the same source node or more usually different destination node. In terms of the WWW, a link can be built between parts of a single page (HTML file), between pages of a single Web document within a Web site (a number of HTML files) or between different Web sites (collections of HTML files). A typical application of this is to have a link anchored at a specific word or phrase in a node, which takes the user to a destination node whenever the word is clicked.

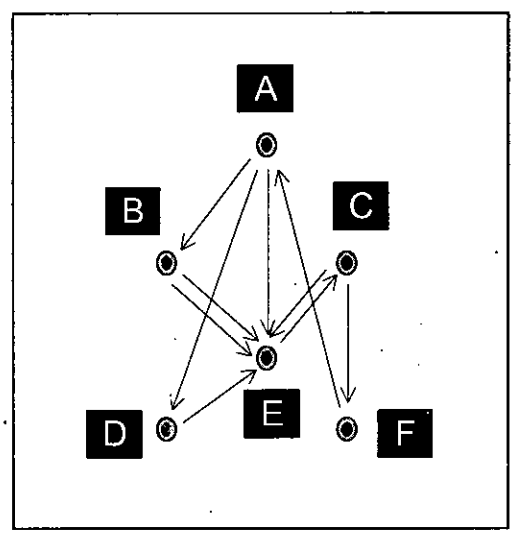


Figure 3.1: A directed graph representation of a hypertext document.

As a hypertext document consists of nodes and links between them, a hypertext document can be viewed as a directed graph. As illustrated in figure 3.1 the edges of the graph represent the links between nodes. There may be a number of links between two nodes, and a particular destination node may have several sources.

The structure of a hypertext document is defined by the hyperlink network. The link structure can be linear, hierarchical, fully connected or any combination of these. Several hypertext researchers (Signore 1995; Conklin 1987; Shneiderman and Kearsley 1989) have investigated structural and directional classification of hypertext links.

The research in this thesis is concerned with the semantic affiliation between hypertext nodes rather than the structural or directional nature of the links between nodes. The hypertext link classification used in MLTutor shares some similarities with those of Shneiderman and Kearsley (1989). They categorise hyperlinks according to the information in the destination node, which

may contain:

- A new topic
- Reference document
- Ancillary information - *glossary, footnote* or *annotation*
- Graphical information
- An index or a table of contents
- An executable program

The link categorisation scheme used in MLTutor is discussed fully in Chapter 7. The objective of MLTutor is to provide guidance to hypertext users. This guidance aims to help users to benefit more from the content of the hypertext system by getting the information into a structurally more efficient order. The classification of any particular link within MLTutor determines whether or not it is used in an analysis of a user's browsing activity; this analysis is used to provide the guidance within the system.

3.5 Advantages of hypertext

Although traditional printed text is considered linear, there are a number of non-linear features commonly seen within the printed media. These features include the use of footnotes, parentheses, cross references and references and are often employed by technical authors to allow complex interwoven arguments to be presented clearly. Encyclopaedias and diagnostic manuals, for example, use these techniques and are rarely read in a linear fashion.

In many respects hypertext can be considered an extension of these non-linear features but offering significant other benefits as outlined by Conklin (1987).

Chief among the advantages of hypertext is the ability to easily pursue cross-references within a text. This allows quick and easy access to additional information but just as importantly, and a major benefit over traditional text, references can be retraced easily.

The ability of a hypertext system to 'remember' the path to a document allows a user to have several lines of enquiry open at a particular time; this is often referred to as task stacking. From a particular point within a document two ways forward may seem appropriate. The user may pursue one of these and subsequently decide the other was more appropriate. The enquiry can be quickly unwound to the original point of divergence and the alternative path taken.

This capability is not readily available with printed documents and a reader has the overhead of having to remember how he got to his current document. This is more of an issue when references refer to additional volumes, which must be retrieved from a library shelf.

As hypertext is an electronic medium and hyperlinks can span documents, the need to access physical volumes is removed. Although this is an advantage for all electronically stored documents, the ability to drill down into a document and backtrack through the references is unique to hypertext systems.

A consequence of this feature is that speed of information access is increased with hypertext. This can be further enhanced by the provision of search facilities that can search content, as with traditional electronic documents, and links across the entire hyperspace. Additional processing can be provided to sort and order links to provide other user benefits. This can be particularly beneficial when links point to non-textual components of a hypermedia system such as images or sound where the only clue to the content might be the link description.

When a hypertext document is constructed, choices about the structure of the text are made by the author who may allow access to a particular piece of information from a number of sources. This facilitates reuse of document fragments and leads to a consistency of information. For example, if a piece of information needs to be updated and is stored in a single fragment referenced from various places, updating the single fragment updates the whole document, removing the need to search for every occurrence. Additionally, as references are embedded in the text, moving the text to another location or even document takes the reference with it, again maintaining consistency.

The fact that a piece of text can be accessed from various sources means that it is not necessarily possible to know what the reader was looking at prior to selecting a link to a particular piece of information. It is thus generally difficult to have a paragraph in a source document continue in a destination document. Far from being a disadvantage this encourages concepts or topics to be modularised. Thus, a particular node is associated with a topic or concept, which usually has a strong real world mapping. In many respects there is an overlap here with the semantic networks of AI research (Conklin 1987).

Additional benefits may also be realised if a hypertext system allows a user to amend or extend the structure of the hypertext themselves. There are many advantages to allowing this flexibility, as an author may not be able to anticipate every user's need.

One way of introducing this flexibility is to allow users to annotate or extend the hypertext content themselves. These may be personal comments that attach to, but do not amend, the original text. This will allow users to have their own set, or multiple sets, of personal comments on a text. For example in a hotel guide a user could add comments on the hotel restaurant if they are not included in the original guide or add personal comments about the quality of service. Extending this concept, annotations could be shared with other users allowing for collaboration on a project.

In addition to allowing content annotation, a hypertext system could allow for a restructuring of content. The MLTutor prototype described fully in Chapter 6 introduces a system-generated method for restructuring a hypertext document. MLTutor generates a dynamic hypertext structure between the Website documents contained within the system and allows jumps to documents within the system that are not directly linked to each other. This dynamically created structure is implemented as the *suggestion list* of MLTutor, which assists users by displaying hypertext pages relevant to their browsing activity.

3.6 Problems with hypertext

Experience with the printed media stretches back for centuries; hypertext by comparison is a relatively new medium and has not yet received wide exposure. Books are usually read sequentially but from a hypertext page it is often possible to access many pages next. This is, of course, a simplification, since it is quite possible to access a book via an index or to flit from topic to topic. However, printed text is essentially linear and the problems cited against hypertext are not often levelled at printed material.

Hypertext systems permit non-sequential, user-driven access to information and, as already outlined in the previous section, offer many advantages over plain text. However, the powerful flexibility of hypertext can result in problems. It is all too easy for a hypertext user to be lured off course; once off course, irrelevant information and further paths can lead to user *disorientation* (Conklin 1987) and *information overload* (Nielsen 1990). The majority of these problems are the result of poor design (Theng 1997; Benyon *et al* 1997).

The three most common causes of *user disorientation*, often referred to as the *lost in hyperspace* problem, are outlined by Elm and Woods (1985) as follows:

- ***Users not knowing where to go next***
- ***Users knowing where to go but not knowing how to go there***
- ***Users not knowing where they are within the overall structure***

The *information overload* problem occurs when a user is swamped with details that may not be relevant to current needs or information is provided in an unstructured manner (Theng 1997).

An attempt to overcome these problems with hypertext has led to the development of a number of navigational support tools. These navigational tools have specifically focused on minimising the likely occurrence of spatial disorientation within a hypertext system. However, they have not specifically aimed to provide additional facilities to assist a user in the execution of a particular task.

De la Passardiere and Dufresne (1992) divide navigational support tools into the following three main categories:

1. Punctual tools support browsing activity within a hypertext system. Examples include labelled links, icons and buttons such as *next*, *previous*, *first*, *last*, *home* or *help*. A link is the commonest way to jump to another hypertext page. The anchor of a link can be words, phrases (Akscyn *et al* 1988) or icons (Goodman 1988) and in some hypertext system these provide extra information about the destination node. Implementations of this tool may vary; when the cursor is moved over an anchor, a brief description of the destination node can be given. Most web browsers support this facility by displaying the full URL (Uniform Resource Location) address of the destination node as soon as the cursor is moved over an anchor.

2. Structural navigational tools can be particularly useful if the hyperspace is quite large. Structural tools aim to give hypertext users information on the organisation and structure of the hypertext documents within the system and include *guided tours* (Trigg 1988), *overview maps* (Utting and Yankelovich 1989) and *fisheye views* (Furnas 1986).

   The *guided tour* is a facility that aims to guide users without requiring too much input and can be valuable for novice users. However, guided tours cannot be the only navigational facility available in a hypertext system as the true purpose of hypertext is to create an open exploratory environment (Nielsen 1990).

Overview maps and *fisheye views* are also structural tools. Overview diagrams can help a user to locate themselves and trace their movements within the information space. They make hypertext links between nodes visible to the users and can be *global* or *local*. Local overview maps provide an overview of the current node and show all the nodes to which it is linked. A global map will cover an entire hyperspace.

Fisheye views show the entire information space, but concentrate on nodes, which are closer to the current node. The closeness of a node to the current node means more detailed information is provided to the user. Similarity between, or closeness of hypertext nodes is often measured in terms of the length of the access path between nodes. For example two hypertext pages which are three links apart from each other are seen as much further apart than two nodes which are directly connected by a link. In other words, measuring the distance between hypertext nodes is based on the topology of the hyperspace rather than content (Dieberger 1994).

3. Historical tools allow direct access to hypertext nodes which have already been visited or specifically marked by a user. The most commonly known historical tools are *bookmarks* and *history lists*. These features are commonly available in WWW browsers.

   A history list facility provides an overview of the hypertext pages visited; for example, a list of all URL's or the title of pages visited may help users to locate where they are within the hyperspace.

   Bookmarks allow a user to mark a node by adding the title of the page to a list. This facility provides a direct link to the bookmarked page at any time.

Although the problems with hypertext can be exacerbated by poor system design, they are also inherent within the hypertext model and result directly from the enhanced freedom of access provided. Designing any document either in electronic or printed-format is a complicated task. For a hypertext document, structure definition becomes even more complicated and there are no globally accepted standards for the design of hypertext documents.

Nonetheless, an important challenge that a hypertext system designer faces is to build structures that match the way that a user wants to read a document, in order to complete a task. Woodhead (1991) suggests that an author's document structure may not suit all users and sometimes even none. At a point in time, a user may be interested in a particular view of a document and a

different view of the same document at another time. Even a good match between a user's needs and that provided by the system may not last very long. Users' tasks change over time and their expectations from the systems follow. A system that is adaptable to user needs is the ultimate solution. Such a system would be powerful but difficult to achieve.

In recent years, researchers have been exploring the viability of building dynamic and intelligent systems to address these concerns. Stotts and Furuta (1991) propose a two layer approach to achieve this; the first layer is formed from the fixed information structure built by the hypertext author and the second layer is a flexible structure that is superimposed on the fixed layer.

In an approach similar to Stotts and Furuta's, MLTutor (see Chapter 6 and 7) generates a dynamic hypertext structure between the Website documents contained within the system. This dynamically built structure aims to assist users in their search for information.

3.7 Conclusion

Most hypertext systems are based on a static and explicit model of hypertext. In other words, the declaration of nodes, links and annotations made during the design process are fixed.

There are many advantages associated with this fixed hypertext format, but, as described in this chapter, there are also problems. In order to address these problems a number of navigational support tools have been developed. These aim to help users while navigating through hypertext pages but do not explicitly help users to achieve a particular goal or find relevant information.

As an alternative, the research within this thesis proposes a structure independent support tool which does assist with the search for information. The tool is adaptive to changing requirements; research on adaptive hypertext relating to the development of this tool is described in the next chapter.

Chapter 4

Adaptive Hypertext

4.1 Introduction

The advantages of hypertext technology based systems were presented in Chapter 3. One-type-suits-all hypertext systems have many advantages over paper based representations. In particular, they offer hypertext authors flexible ways of presenting information, and system users have greater navigational freedom.

However, there are problems associated with this increased flexibility and research into the two principal problems, the *lost in hyperspace* and the *information overload* problems, were discussed in the previous chapter. This chapter aims to revisit these problems and present adaptive hypertext methods and techniques for their solutions. Following the theoretical discussion of the techniques employed by adaptive hypertext systems, the implementation of these techniques in number of recent systems are reviewed.

With the advent of the WWW, educational hypertext has a new medium which provides an opportunity to offer global distance learning (Laurillard *et al* 1998; Dwyer *et al* 1995). The volume and variety of educational applications on the WWW is increasing immensely but without an adaptation mechanism these systems have often failed to provide users with information matching their needs (Brusilovsky *et al* 1996a).

4.2 What is adaptive hypertext?

Adaptive hypertext can be viewed as the historical successor of hypertext as shown in figure 4.1. The need to overcome problems with non-adaptive hypertext systems has motivated the evolution of adaptive hypertext and hypermedia research (Brusilovsky 1996).

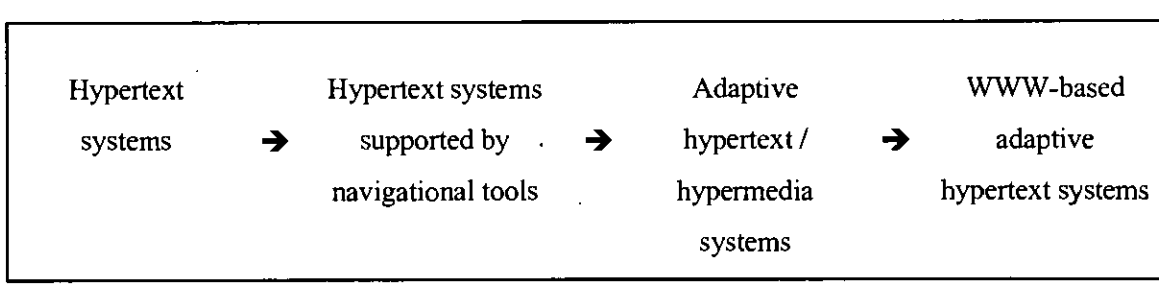


Figure 4.1: The evolution of hypertext.

Adaptive hypertext technology based systems aim to support users by tailoring the system and augmenting the delivery of information. An effective adaptive hypertext system will be capable

of filtering out details that are outside a user's current field of interest or beyond their level of comprehension. In effect, adaptation controls the size of the hyperspace available to a user at a particular point in time.

4.3 Why is adaptive hypertext necessary?

The primary goal of an adaptive hypertext system is to tailor the hypertext system to the specific needs or preferences of a user. Early studies in the field of intelligent tutoring systems (ITSs) have indicated that there is a need to understand a user's specific information requirements and background knowledge in order to provide effective tutoring. An effective tutoring system will present material to a student at a level that matches their abilities and will adapt as the student's knowledge develops. A similar control of information within a hypertext system offers the prospects of addressing both the *lost in hyperspace* (Theng 1997; Boyle and Encarnacion 1994) and the *information overload* problems (Hook 1997).

4.3.1 A solution to the problems with hypertext

A hypertext system allows users more control during a search for information by allowing topics to be freely explored as needed (within the constraints of the system design). Although this flexibility is powerful it may result in the information overload and disorientation problems which led to the development of tools such as overview maps, local maps and filters which were discussed in Chapter 3. These navigational tools have specifically focused on minimising spatial disorientation within a hypertext system. Adaptive hypertext attempts an alternative approach to these problems by controlling what is made available and / or the format in which information is presented.

4.3.2 A solution for differing user preferences and abilities

Not all users of a hypertext system will necessarily have the same goals or abilities. Some users may be experts looking for specific information, others may have more general requirements, while others may be novices trying to find basic information. Kobsa *et al* (1994) propose adaptation of hypertext as a solution for catering with different user requirements, interaction styles, background knowledge, and cognitive characteristics. An adaptive hypertext system tries to identify a user's goals and adapt to these. As suggested by Kobsa *et al* (1994) novice users can be catered for by a specialised or simplified presentation of information and an expert's needs can be addressed by enhancing the navigation.

4.4 How can hypertext be adapted?

Adaptation is a powerful way of augmenting the functionality of a hypertext system. There are two main components of a hypertext system that can be adapted; these are hypertext links and information contained in the nodes. Adaptation of hypertext links mainly affects navigation within a hypertext system whereas adaptation of the nodes themselves affects the presentation of information. These two forms of adaptation are usually referred to as *navigational* and *presentational* (Brusilovsky 1994a; 1996). These adaptation techniques are presented in the following sections.

4.4.1 Presentational adaptation

Presentational adaptation aims to adapt the information being presented to the user with a view to hiding details not of current interest. Techniques used to accomplish presentational adaptation are *conditional text*, *stretchtext*, *page variants*, *fragment variants* and *frame-based* representations.

Conditional text: With this technique a concept is divided into chunks of text. Each chunk of text is associated with a condition indicating which type of user should be presented with the information chunk. Expert and novice users may be presented with different chunks for the same concept as illustrated in figure 4.2.

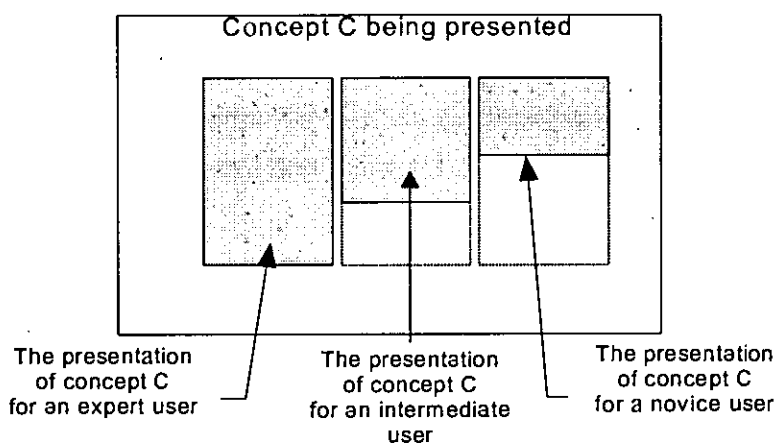


Figure 4.2: A conditional text example.

This figure suggests that an expert is presented with more information than a novice; however this may not be the case in all systems. For example, in an educational system a novice user may require more information than an experienced user.

Stretchtext: This is a widely used technique (Hook 1997; Boyle and Encarnacion 1994) to give users additional explanation related to the current topic. Instead of retrieving a new page, clicking on an active link or hotword results in additional text being displayed in a pop-up window. An example is illustrated in figure 4.3; clicking on the hotword 'a shortcut menu' provides a user with additional information.

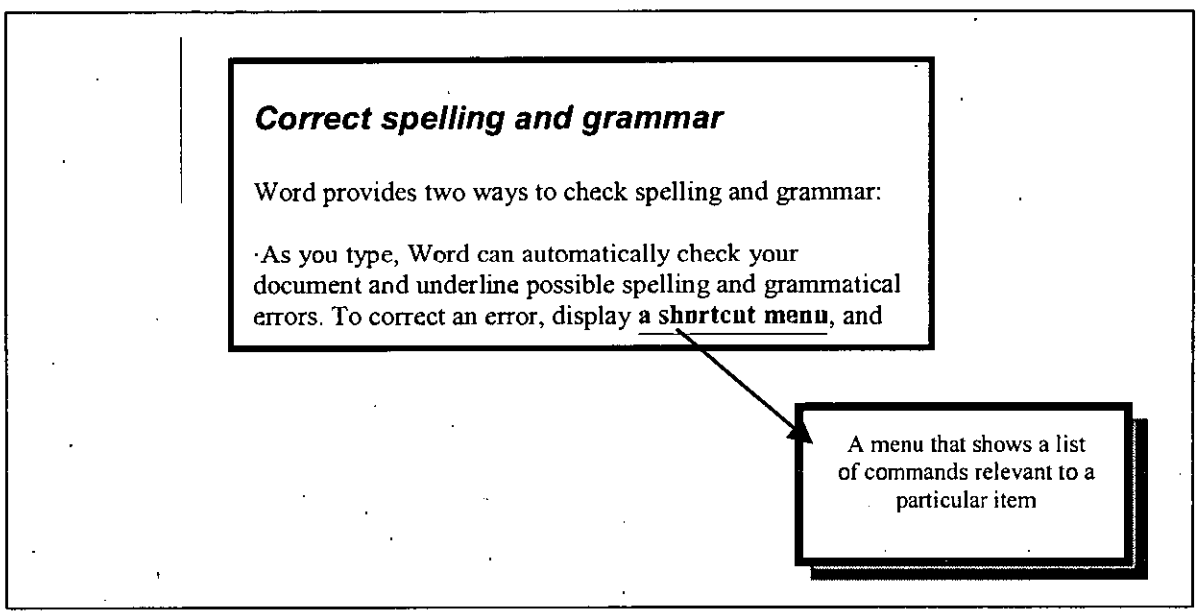


Figure 4.3: A stretchtext example.

Page variants: With this technique two or more variants of the pages associated with a concept are prepared as shown is figure 4.4. As in Metadoc (Boyle and Encarnacion 1994), each variant of the page presents information at a different level or in a different style and the system selects the most appropriate page variant for the user.

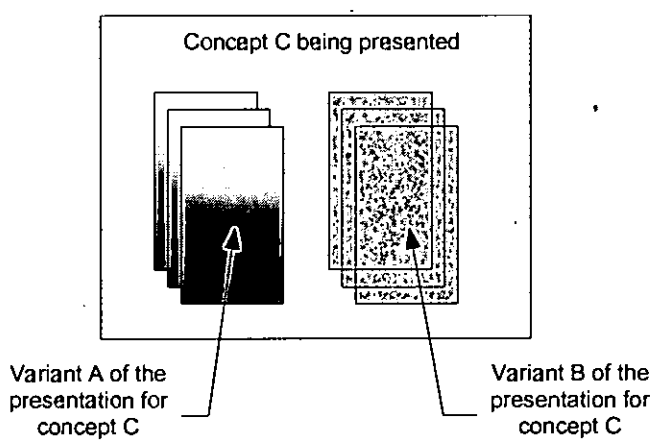


Figure 4.4: Two presentations of a particular concept.

Fragment variants: This is a more fine-grained implementation of the page variant technique. With this technique as illustrated in figure 4.5, each page is broken into a number of fragments and a number of variants of each fragment are prepared. As with the page variant technique the system selects the most appropriate version of each fragment to present to the user.

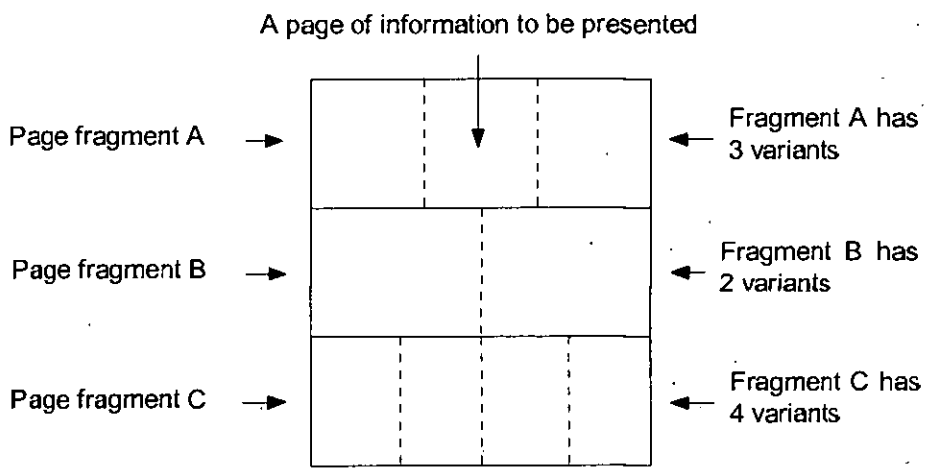


Figure 4.5: Multiple variants of fragments on a page.

Frame-based: When using the frame based technique a concept is represented in the form of a frame structure. Each slot of the frame contains a variant of the same concept and also can be linked to other frames. A system using this technique also embodies a set of rules to calculate the most appropriate slot to be presented to a specific user.

The adaptation techniques discussed above are concerned with adaptation of content rather than link-level adaptation (Brusilovsky 1996). These methods focus on adapting the content of information to users in a way that it is assumed to be more suitable to their current knowledge, goals and requirements. The adaptation available in MLTutor (see chapters 6 and 7) is navigational rather than presentational. Navigational adaptation is discussed below.

4.4.2 Navigational adaptation

Navigational adaptation is the most widely used approach (Brusilovsky 1996) to support hypertext adaptation. In recent studies including InterBook (Brusilovsky and Eklund 1999; Eklund *et al* 1997), ELM-ART (Brusilovsky *et al* 1996a; Brusilovsky *et al* 1996b; Schwarz *et al* 1996) and ELM-ART II (Weber and Specht 1997), the navigational adaptation techniques are referred to as Adaptive Navigation Support (ANS) techniques.

The aim of navigational adaptation is to assist users by manipulating the navigational aids (links, labels, hotwords) within the system suggesting appropriate directions to take. It is proposed that navigational adaptation can overcome the problem of user disorientation and also information overload by constraining the exploration of the entire hyperspace. Navigational adaptation can be applied both globally and locally. When applied locally the scope of any hypertext searches are locally restricted. A global approach is required if the user's goal can be found in many locations and the shortest path to it is required. The scope of a global approach is the whole hypertext system and can be as extensive as the WWW.

Changing hypertext boundaries by link manipulation, by either shrinking or enlarging the hyperspace, provides navigational adaptation. Link manipulation can be used to suggest the most relevant link to follow, links can also be activated, de-activated or dynamically added. Techniques to accomplish link manipulation can be grouped into five categories as follows:

- *Annotation* as in ELM-ART II, AST and ISIS-Tutor
- *Ordering or link sorting* as in HYPERFLEX and Hypadapter
- *Direct guidance* as in WebWatcher, ISIS-Tutor, AST, and ELM-ART
- *Hiding* as in PUSH and Hypadapter
- *Mapping as in* HYPERCASE

Annotation

Adaptive annotation is a popular ANS technique (Brusilovsky 1996). With this technology links are enriched with extra comments or visual cues. The use of annotation links aims to provide users with more information about the destination of a link prior to selection. Annotations can take the form of text and icons or can be encoded by colours, different font sizes or typefaces.

Brusilovsky and Pesin (1995) suggest that ANS may help users by providing them with more appropriate mental maps to follow and so reduce floundering in the hyperspace of information. This may be beneficial in terms of adaptive curriculum sequencing in educational. There are three common styles of annotation as follows:

- The history-based annotation technique is not a new technology. It has been widely used in WWW browsers to indicate whether a link has been visited or not.

- The knowledge-based annotation technique is used to identify a user's knowledge on a particular topic contained in a node. Knowledge-based annotations are used in MANUAL

EXCEL (de la Passardiere and Dufresne 1992) to denote links as *not-known*, *in-work* and *well-learned*.

- The prerequisite-based annotation technique determines the educational pre-requisites of each hypertext page, based on the knowledge level of a user. In ELM-ART (Brusilovsky, *et al* 1996a; 1996b), this type of annotation technique is used in the form of a 'HELP' button. When a user makes a help request, the system provides the user with information on the background relevant to the concept being learned.

Ordering or link sorting

This technique sorts or reorders the links on a specific page or topic according to a user model. The applicability of adaptive sorting in hypertext systems is very limited and can never be used with contextual links and maps (Brusilovsky 1996). However, this method has been used in a number of information retrieval systems such as HYPERFLEX (Kaplan *et al* 1993) and the POP hypertext help system (Hook *et al* 1996). A link ordering technique is used by MLTutor, which is described fully in Chapter 6.

Direct guidance

This is the simplest adaptive navigational aid. With this technique the system indicates to the user the best node to visit next as in WebWatcher (Armstrong *et al* 1995), or the next node as in ISIS-Tutor (Brusilovsky and Pesin 1994). Usually, direct guidance does not give the user the flexibility of ignoring the system's suggestion (Brusilovsky 1996) and is often used to provide curriculum sequencing in educational systems. In ELM-ART II (Schwarz and Weber 1997), the 'NEXT' button is used to provide direct guidance.

Hiding

This technique controls access to information by hiding or disabling links to pages, which are irrelevant to the user's requirements. This reduces the possibility of cognitive overload by controlling the navigable hyperspace. Because hiding allows for a gradual exposure of the hyperspace, it can be used effectively in educational hypermedia systems such as ISIS-Tutor (Brusilovsky and Pesin 1994). This technique can be applied to non-contextual and contextual components such as system links, indexes, maps, menus and icons. It is used in the PUSH (Hook *et al* 1996) system.

Mapping

A map allows a user to understand the overall structure of the hyperspace and also to locate

themselves within it. The hiding, annotation and direct guidance techniques can be used to enhance a map. The map adaptation technique employed in HYPERCASE (Micarelli and Sciarrone 1996) assists users when they require help by showing them where they are within the hyperspace by drawing global or local area maps.

Navigational adaptation methods have been developed with the intention of solving various problems related to navigation within a large information space. A number of adaptation techniques can be used both locally and globally. The most widely used adaptive navigational techniques which support both the local and global approach are direct guidance and link sorting. The adaptation within MLTutor, described in Chapter 6, is navigational and global, and takes the form of an annotated suggestion list.

4.5 Typical adaptation in adaptive hypertext systems

Brusilovsky (1996) describes the adaptation cycle of an adaptive hypertext system in terms of a three stage process as illustrated in figure 4.6. This is a dynamic process which takes place during system execution and allows the system to adapt to a user's requirements.

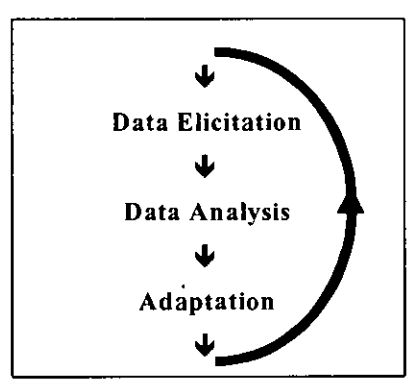


Figure 4.6: The typical adaptation cycle of an adaptive hypertext system.

The first phase of the adaptation process is data elicitation and is concerned with gathering information about users. The aim is to determine the style or the level of information to be delivered to users. In general, gathering information about a user's area of interest can be accomplished initially or during the learning activity. In adaptive systems, as discussed in §4.8, the most popularly applied methods are *initial interviews* (Brusilovsky and Pesin 1994; Beaumont 1994), *question and answer analysis* (Weber and Specht 1997; Brusilovsky *et al* 1996b; Beaumont 1994; Kobsa *et al* 1994), *user relevance feedback* (Kaplan *et al* 1993; Mathe and Chen 1994) and *storing user browsing patterns* (Kobsa et al 1994; McEneaney 1999, Hirashime *et al* 1998).

Brusilovsky (1996) suggests that an ideal system would be able to collect data describing a user's activity, process the data to build a user model and provide adaptation without requiring any external data. He comments on an ideal system as follows, " ... while the user is simply working in an application system, the adaptation component watches what the user is doing, collects the data describing user's activity, processes these data to build the user model, then provides an adaptation. Unfortunately, such an ideal situation is very rarely met in adaptive hypermedia systems...". Brusilovsky (1996) concludes that almost all adaptive hypertext systems rely on an external source of information that is typically provided by the users themselves via *question and answer* sessions, *initial interviews* or *relevance feedback* techniques.

The feasibility of providing adaptation based on a user's browsing patterns has been investigated by a number of researchers (Sun *et al* 1995, Kobsa *et al* 1994). Beaumont (1994) argues that the bandwidth of the information contained in a user's browsing pattern might be too narrow to elicit sufficient information about the user's interests. In contrast, Sun *et al* (1995) argue that since one of the advantages of a hypertext system is to provide users with a high degree of navigational freedom to search for information, the browsing pattern of a user is a significant information source.

While browsing patterns have been investigated by authors such as Sun *et al* (1995) and McEneaney (1999), little work has been done on applying machine learning techniques to dynamically build user profiles in the field of adaptive hypertext.

In MLTutor, the navigational steps taken by users are the sole information source analysed to understand motivation. This approach is the most direct and unobtrusive way of gathering information about a user's interests during a task-oriented information search. Furthermore, this approach allows adaptation to be provided without disturbing the natural flow of user navigation.

The second phase of the adaptation process described by Brusilovsky (1996) is data analysis. During the analysis phase the data collected on each user is processed to build or maintain a user model. Frame-based as in ELM-ART II (Weber and Specht 1997), and rule-based knowledge representations of the collected data are commonly utilised techniques.

In the research presented in this thesis, the feasibility of applying a machine learning approach to user modelling has been investigated. The machine learning component embedded in MLTutor integrates a number of machine learning techniques to process a user's browsing

history. The result of this processing is a set of dynamically created rules which describe the users interests. These rules can be viewed as the foundation for an individualised user model which can be utilised to provide adaptation either navigationally or presentationally. The machine learning techniques employed by MLTutor are discussed fully in Chapter 5.

The third phase of the adaptation cycle is utilisation of the analysis to provide user focused adaptation. The most popularly used adaptation techniques, as discussed in the preceding sections, are adaptive link annotation as in ELM-ART II (Weber and Specht 1997), AST (Specht *et al* 1997) and ISIS-Tutor (Brusilovsky and Pesin 1995) and stretchtext as in ANATOM-TUTOR (Beaumont 1994) and Metadoc (Boyle and Encarnacion 1994).

MLTutor introduces navigational adaptation into a Web-based information system via a suggestion list. The suggestion list is constructed using the rules dynamically generated by the machine learning component of the system and lists pages related to those accessed by the user. The implementation of the suggestion list restructures access to information as opposed to completely removing it and also introduces a basic adaptive link annotation feature.

4.6 Student models in adaptive hypertext systems

Eliciting information about a user, based on interaction with the system, and anticipating a user's requirements is a key concern in an adaptive hypertext system (AHS). As mentioned above methods used to gather information, either directly or indirectly, about a user can be categorised into four main styles: *question and answer, initial interviews, relevance feedback* and *browsing behaviour*. The information from these sources is available to build and maintain a model of a user within the system.

Building a user model based on *browsing behaviour* alone is a challenging objective. A common alternative has been to use stereotypical user profiles in the user model and map an actual user onto one of the stereotypes after an initial interview as in as in ANATOM-TUTOR (Beaumont 1994), KH-AHS (Kobsa *et al* 1994), Hypadapter (Hohl *et al* 1996) and AST (Specht *et al* 1997). However, in order to provide continual adaptation within the system, an additional source of information is needed to reflect the ongoing state of the user within the model.

User modelling issues have been investigated by researchers in the field of ITSs (Kobsa and Wahlster 1988) where the term student model is preferred. Student models are used in intelligent tutoring systems (ITSs) in order to guide students towards a better understanding of

the teaching material and to tailor the material to the student. From the perspective of educational value, ITSs have often been criticised for carrying out teaching or remediation based on the embedded assumptions in their student model which may be invalid (Kass 1988; Sparck Jones 1988). On the other hand, non-adaptive hypertext systems have been criticised for the lack of structural and instructional guidance within them. Early attempts to address these issues in non-adaptive hypertext did not involve the use of a user model within the system. These early solutions were discussed in Chapter 3.

In §4.8.1, a number of hypertext systems are reviewed with an emphasis on the user / student model employed and how these models are used in the adaptation process.

4.7 Opportunities for AI in adaptive hypertext development

The aim of combining hypertext research with artificial intelligence (AI) is to build 'intelligent' features that are capable of turning the passive collection of information into an intelligent hypertext system. Early hypertext systems were limited in terms of making inferences about hypertext links or information contained in hypertext nodes.

Shneiderman and Kearsley (1989) suggest that AI techniques may contribute to hypertext development in two main areas; *truth maintenance* and *learning*. In their view, a system with a truth maintenance capability should be able to add, modify and delete currently presented information if necessary. A system with a learning capability should be able to record paths that users take and restructure the links to facilitate rapid navigation or to inform users of shortcuts. Today, researchers in the domain of adaptive hypermedia prefer the terms *navigational* adaptation and *presentational* adaptation (Brusilovsky 1996).

4.8 Major developments in adaptive hypertext research

An adaptive hypertext system is suited to situations where it is expected that users will have various levels of background knowledge, different requirements, or expectations. There are three principal areas where adaptive hypertext has been employed. These are on-line help systems, document retrieval systems and tutoring systems.

With the advent of the WWW, it is now relatively easy to make a massive amount of information available to a very large audience. Consequently, extracting information that is

really relevant from the masses of information available becomes harder. In a similar way to intelligent help and tutoring systems, information retrieval systems aim to understand information requirements as accurately as possible and deliver the required information. Among these different fields of study there are shared objectives which, in recent years, has drawn them closer together leading to a sharing of technologies.

A number of adaptive hypertext systems are discussed below with an emphasis on the user / student models within them. The adaptation that each system incorporates is described along with the various methods that are employed to update the user / student models that the systems contain.

4.8.1 A survey of adaptive hypertext systems

ELM-ART (Brusilovsky *et al* 1996a; 1996b)

ELM-ART (Brusilovsky *et al* 1996a; 1996b) is the WWW version of ELM-PE (Weber and Mollenberg 1994) which was developed to teach the programming language LISP to beginners. ELM-ART (Adaptive Remote Tutor) is an adaptive electronic textbook which uses adaptive annotation links (see §4.4.2). Prerequisite-based help and visual cues such as icons, fonts and colours are used. In this system, the prerequisite-based help has been implemented in the form of a button. When a user makes a help request, the system provides the user with information related to the background concepts of the current topic. In the student model, each hypertext page is categorised into educational states *learned, ready to be learned* and *not ready to be learned* which means that prerequisite knowledge has not been learned yet. This information is reflected back to a user by changing the colour of the links on pages. The colour changing scheme is as follows: Green coloured links mean ready to be learned; red coloured links mean not ready to be learned; yellow colour links are ready to be visited but not recommended by the system and white coloured links mean known. The user model is an overlay model; for each concept of the domain, the model stores a value reflecting the student's knowledge level for the concept.

ELM-ART II (Weber and Specht 1997)

ELM-ART II is the latest version of the ELM-ART system, which has been re-developed using the electronic textbook-authoring tool InterBook (Brusilovsky *et al.* 1996). The domain knowledge is split into different categories such as lessons, sections, subsections and terminal pages. Information in each category is stored in frames. Each frame contains a number of slots that hold information such as the text to be presented and information about the relation between the frame and others. Beyond these static slots, each individual learner model contains

dynamic slots which are updated during the learner's interaction with the system. Information in the dynamic slots is used to annotate links to provide the learner with an optimal path. As in ELM-ART, ELM-ART II uses visual cues to provide navigational adaptation within the system. Prerequisite-based annotation is used; next to each topic or on top of each page, green, red, yellow and orange colour balls are used to guide users. This system also features curriculum sequencing. If a user clicks on a button called 'NEXT' the system selects the best next step for that particular user to follow.

AST (Specht *et al* 1997)

AST (Adaptive Statistic Tutor) is a tutoring system on the WWW, which aims to generate individualised courseware. The system updates a probabilistic learner model based on a user's interaction with the system. A new user has to answer questions about their background knowledge and preferences. Users can pick one of the system's teaching strategies, such as learning by examples, learning by reading text or learning by doing, and can also define the level of the concept. Each concept can be presented at three levels, the first level contains basic information, the second gives more detailed information and the third gives advanced information. This is an example of the presentational adaptation employed in AST. Navigational adaptation is provided by AST in a similar way to ELM-ART – that is, by link annotation, where the state of each link is represented with a coloured ball. If a user asks which topic in the curriculum is to be read next, the system computes the next best topic depending on information stored in the probabilistic overlay model.

ISIS-Tutor (Brusilovsky and Pesin 1994)

The ISIS-Tutor is a hypermedia-based intelligent learning environment. The system integrates a tutoring component which presents problems to the student, a hypertext component which allows for user driven knowledge acquisition, and a learning environment which allows for experimentation with the taught material.

The design of the system is based on a domain model of the teaching material, represented as a directed graph of concepts indicating which concepts are preconditions of others. The structure of the hypertext component of the system is based on this network, which also forms the basis of the student model. This is an overlay model, in which the student's current understanding is represented as a subset of the domain knowledge. The student model records an integer for each concept within the domain model network which, via the mapping to the hypertext, corresponds to a node.

These integer values are updated by an evaluation module which analyses the student's problem solving ability within the tutoring component. The tutoring component selects an appropriate teaching strategy based on the values in the student model which constrains the navigation within the hypertext. The student model is the heart of the system and the various components of the system adapt navigationally and presentationally by comparing the integers stored within it against threshold levels, which may differ for the various system components. Within the hypermedia component four states are recognised allowing nodes to be marked as not ready to be learned, ready to be learned, in work or learned. This is an example of the prerequisite-based annotation.

Although this simple approach to student modelling allows all the components of the system to access the student model, and so to adapt, it was reported by Brusilovsky and Pesin (1994) that co-ordinating updates from the various components was less successful. It also became apparent that, while the information stored was highly suitable for the tutoring component, it was less so for the others. In particular, the hypertext needed to know how often a particular hypertext page had been presented.

A counter in the student model was used to store this information, however, as the simple overlay student model in the ISIS-Tutor had a single counter for each concept, this value was subsequently over written by other components within the system. This led to the adoption of a more advanced student model that incorporates projections from a central student model for the various components of the system (Brusilovsky 1994b). A projection is created from the central student model by projection rules that present student information in a form necessary for the component to adapt. The central student model stores partly processed information about the student to avoid loss of information that a particular component might need.

Metadoc (Boyle and Encarnacion 1994)

Metadoc is an adaptive information presentation system. In the system there are four versions of each page and the stretchtext on each version of a page differs. Metadoc employs four different user stereotypes, which are novice, beginner, intermediate and expert. The classification of the user is based on their knowledge of Unix/AIX and general computer concepts. This system aims to automatically adapt the hypertext document to the ability of the reader. Based on its user model the system alters the amount of information presented to the user by automatically stretching or contracting the text in a node. The stretching and collapsing can be overridden via mouse clicks or using detail buttons.

Hypadapter (Hohl *et al* 1996)

The Hypadapter system uses a combination of presentational and navigational techniques. This system supports self-initiated, goal-oriented exploratory learning in the Common Lisp programming language. Within the system, individualised adaptation is initiated by assessing users with a questionnaire. Hypadapter employs four user types, which are novice, beginner, intermediate and expert. Like ELM-ART II, this system uses a frame-based knowledge representation. Hypadapter uses an ANS technique for sorting fragments of information relevant to a user's knowledge. Hiding is another technique used within the system. Unsuitable links are hidden behind a special icon, however, these can be activated by a user clicking on the icon.

ANATOM-TUTOR (Beaumont 1994)

ANATOM-TUTOR is an intelligent tutoring system which integrates an educational hypertext. As with ISIS-Tutor there are three modes of operation for the system. In this case, they are question mode which interrogates the student, browsing mode which allows self directed access to the domain knowledge, and hypertext mode where presentation is guided.

In contrast to ISIS-Tutor, where all information is gathered during interaction, users of ANATOM-TUTOR undergo a brief enrolment procedure in which their current knowledge is assessed. At this stage, only key concepts are covered and stereotype information is applied to classify the student. The student's answers are matched against sets of stereotypical answers and the student is thereby allocated to a particular stereotype. This information is recorded in a rule based student model, which is consulted by the tutoring module (the didactic module) to adapt its presentation. This model is updated as the student gives answers to questions posed in question mode. Navigational and presentational adaptation are both features of the system and take place on the basis of details recorded in the student model. Presentational adaptation is accomplished by using the fragment variant technique.

Relevance Network (Mathe & Chen 1994)

The user's view of the relevance of information to their goals is at the heart of Mathe & Chen's Relevance Network. This is an adaptive information retrieval system that can be used within a hypertext environment. As a user views documents, they are able to indicate if a document is relevant or not to their current goals. These goals are specified, along with user specific information, in a profile. This information is stored within a relevance network, which relates a generalisation of the query used to search for the document (if a query has been used) to the currently selected profile and the retrieved document.

Whenever a user requests information, the relevance network can be checked, if the user wishes, to see what information was marked as relevant for previously executed similar queries. These may be queries executed by the current or past users since the system allows for profile sharing. The user can choose which, of the documents listed as being relevant, to access. This information, as with the HYPERFLEX system discussed later in this section, is used to guide the navigational adaptation available within the system. There is no presentational adaptation.

KN-AHS (Kobsa *et al* 1994)

KN-AHS is an adaptive hypertext browsing system, rather than a tutoring system. Student modelling facilities are provided via the BGP-MS user modelling shell system (Kobsa 1990). This is an off-the-shelf package that uses stereotypes, arranged into a hierarchy. The stereotypes implemented within KN-AHS are an 'any person' stereotype which holds general information, and more specialised stereotypes such as 'hypertext user' or 'PC user'.

The initial stereotype for a user is selected on the basis of an initial interview. The user model is subsequently updated according to a set of heuristics on the basis of selections the user makes within the hypertext. For example, if a user deselects an item then familiarity is assumed. If an explanation or definition is requested then unfamiliarity is assumed. The adaptation in KN-AHS is presentational and occurs as a new page is prepared for presentation. If the student model indicates unfamiliarity with the associated concept, an explanation is inserted, whereas if the user is believed to be unfamiliar with the concept then additional information can be provided.

HYPERFLEX (Kaplan *et al* 1993)

Like KN-AHS, the HYPERFLEX system is an adaptive hypertext browser. However, in this case the adaptation is navigational, rather than presentational. A distinguishing feature of this system is that the user can always access all topics within the system as a full list of topics is always available. The adaptive component of the system orders the list such those that topics considered most relevant, to the current goal (if specified) or accessed topics, are placed at the head of the list.

The domain model underlying HYPERFLEX is a fully connected semantic network; that is, every topic can be reached from each topic. This information is encoded as a topic to topic association matrix. The values in this matrix indicate the strength of relationship between topics. In addition, a goal to topic associative matrix exists whose values indicate the relatedness of goals to topics. The value of the matrix weights are set by a user's interaction with the system and so represents the user's view of the domain and forms the user model. The values or weights within the matrices are combined to determine the order of the suggestion list.

Relevance feedback is used to update the associative matrix weights. There are two modes of achieving this. In manual mode the user reorders the suggestion list, moving more relevant topics to the top. In automatic mode, the length of time spent looking at a topic is used as a guide to relevance. However, as Kaplan himself acknowledges, there are significant flaws with this time-based strategy as there is no guarantee that the user is actually reading the topic for the full time that it is displayed.

4.8.2 Summary of adaptation

In summary, the above systems adapt the presentation of, or navigation routes through, information by assessing the cognitive state and intentions of their users. This assessment is based on user input and is variously captured by:

Question and answer analysis (as in ISIS-TUTOR and ANATOM-TUTOR)
Initial interviews (as in ANATOM-TUTOR, KN-AHS, Hypadapter and AST)
Analysing user browsing patterns (as in KN-AHS, ELM-ART and ELM-ART II)
User relevance feedback (as in HYPERFLEX and the Relevance Network)

The adaptation that results from this assessment is varied. Navigational adaptation features prominently in the Relevance Network system and the HYPERFLEX system, where it is implemented as a suggestion list. One clear advantage of adaptation in this form is that it is unobtrusive and can easily be disregarded if not deemed appropriate. However, this approach is limited to navigational adaptation, in a context where the user is knowledgeable enough to recognise from page titles which pages of information are likely to be relevant to them.

In the context of tutoring systems, users are typically unfamiliar with the material, or with at least parts of it, at the beginning of the interaction. One approach to establishing what the user knows is that taken in KN-AHS and ANATOM-TUTOR. Both of these systems use an initial interview to gather information, which is then used to classify users into a number of stereotypes. This technique simplifies system design, but the classification may not be ideal for all users, especially those who are on the borderline between stereotypes. For this reason a mechanism must exist to allow transitions between stereotypes, particularly as users' goals and knowledge change over time. KN-AHS uses user selection to complement information gained during an initial interview, while ANATOM-TUTOR bases updates on the user's answer to questions.

4.9 Conclusion

As with intelligent tutoring systems, an adaptive hypertext system aims to tailor itself in order to meet the needs of users. The research presented in this thesis proposes a novel approach for creating adaptation in a Web-based information system. This approach employs machine learning algorithms to generate personal adaptation based on a history of a user's browsing activity in hypertext. In order to demonstrate the viability of this proposed approach, the MLTutor research prototype has been developed. The hypertext content of MLTutor is a collection of WWW documents, and the machine learning algorithms embedded within the system analyse a user's interaction with the hypertext. MLTutor uses this analysis to generate a list of suggestions. The suggestions are topics within MLTutor and a link ordering technique is provided as in the HYPERFLEX system (Kaplan *et al* 1993). However, in contrast to the HYPERFLEX system, MLTutor does not require any user relevance feedback to assist users. The development of MLTutor is discussed fully in Chapter 6.

In the next chapter an introduction to machine learning concepts and algorithms is presented.

Chapter 5

Machine Learning

5.1 Introduction

This chapter introduces a number of concepts in Machine Learning (ML), a branch of research in the field of Artificial Intelligence (AI). ML algorithms are at the core of the MLTutor system and are responsible for generating the *suggestion list*, which introduces adaptive navigational support into the system.

The application areas of ML, which are closely related to the research undertaken in this thesis are intelligent tutoring systems, adaptive WWW-based educational systems and intelligent personal assistant systems. The main aim of this chapter is to present a number of symbolic ML algorithms along with systems from these application areas which make use of the algorithms.

5.2 Machine Learning

A significant issue with non-intelligent (computer) systems is their inability to cope sensibly with new or unexpected situations. Anticipating every possible scenario is only feasible in restricted environments and not practical for systems expected to operate autonomously in a real world setting.

In order to cope with such situations the ability of a system to learn is seen as a key requirement by *artificial intelligence* researchers. At the present time, computational learning theory in AI research splits roughly into three main areas: Firstly, *symbolic* machine learning that investigates computational learning using symbolic algorithms. Secondly, *connectionism* which investigates computational learning using networks of simple, neuron-like units. Finally, *genetic algorithms* which provide a learning method motivated by an analogy to biological evolution (Goldberg 1989).

Research in the field of symbolic machine learning (ML) has resulted in the development of a wide range of algorithms. Typically, learning in these algorithms is accomplished by searching through a space of possible hypotheses to find an acceptable generalisation of a concept. However, ML algorithms vary in their goals, learning strategies, the knowledge representation languages they employ and the type of training data they use.

ML algorithms that do not require training are referred to as *unsupervised* algorithms e.g. *clustering* and *discovery* algorithms. Those that require training with a set of pre-classified examples are referred to as *supervised* learning algorithms e.g. *decision tree learning* and

version space algorithms. In the following sections a number of supervised and unsupervised concept learning algorithms are described along with various systems that use them.

5.2.1 Decision Tree learning

Concept learning or *Induction* is the task of constructing a class definition. Concept learning algorithms often generate class definitions in the form of *decision trees* which are able to solve difficult problems of practical importance. A decision tree consists of the following and is a representation of a procedure for determining the class of a given instance (Utgoff 1989).

- Leaf or answer nodes that indicate a classification either positive or negative.
- Non leaf or decision nodes which contain an attribute name and branches to other decision trees or leaf nodes, one for each value of the attribute.

The top down induction of decision trees is an approach to decision tree building in which classification starts from a root node and proceeds to generate sub trees until leaf nodes are created. It is possible to categorise conjunctive and disjunctive descriptions of concepts with decision trees and *if-then* rules can easily be lifted from the trees.

5.2.2 The Classification Algorithm

The classification algorithm is a non-incremental, supervised concept learning method that produces a hypothesis in the form of a decision tree. The algorithm accepts a training set of attribute based positive and negative examples of a concept which must all be presented before learning commences, hence the algorithm is non-incremental. The following example applies the classification algorithm to the training set in figure 5.1. In this training set, given by Quinlan (1983), people are described by three attributes which are 'height', 'hair colour' and 'eye colour'. For example Case 1 describes a short person with blond hair colour and brown eyes which is a negative example of the concept being learned. Conversely, Case 3 describes a tall, blond haired person with blue eyes which is a positive example of the concept being learned.

Case	Height	Hair	Eye	Classification
1	short	blond	Brown	-
2	tall	dark	Brown	-
3	tall	blond	Blue	+
4	tall	dark	Blue	-
5	short	dark	Blue	-
6	tall	red	Blue	+
7	tall	blond	Brown	-
8	short	blond	Blue	+

Figure 5.1: An example training set given in Quinlan (1983).

The classification algorithm proceeds by randomly selecting an attribute to add to the decision tree and branches are grown for each possible value of the attribute. The training examples are added to the tree and the classification of the examples at the node checked. If all the training examples at a node are positive or negative the node is labelled with that classification and becomes a leaf node of the tree. Otherwise this process is repeated recursively until all nodes are leaf nodes. The algorithm is summarised as follows:

Input: A set of training instances C

Output: A decision tree

1. If all elements in C are positive then create a 'yes' node and halt.
2. If all elements in C are negative then create a 'no' node and halt.
3. Otherwise
   - Select an attribute A with values $\{A_1, A_2, \ldots A_v\}$
   - Partition C into subsets $C_1, C_2, \ldots C_v$ according to their values on A,
   - Create a branch with A as parent and C_1 etc. as child nodes,
   - Apply the procedure recursively to each child node.

The following example applies the classification algorithm to the training set in figure 5.1. Firstly an attributes is chosen randomly, for example the 'height' attribute, which is made the root node of the tree. There are two possible values for height so two child nodes are created. These are 'short' and 'tall' and a branch is created for each of these. The training cases which have a height value of 'short', in this case 1, 5 and 8, are added the tree under this branch. Similarly the training cases which have a height value of 'tall' are added. The tree at this stage is shown in figure 5.2. The training cases on a branch of the tree are checked to see if they are all

positive examples of the concept, or all negative examples of the concept being learned and if so the branch is labelled '+' or '-' as appropriate. This is not the case here and so the procedure is recursively applied to each branch.

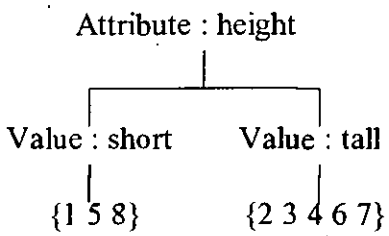


Figure 5.2: The decision tree after the first cycle of the classification algorithm.

The branches are expanded by randomly selecting another attribute, not yet appearing on the path from the root, for example 'hair'. The branches are labelled with the attribute name and legs grown for all the different values possible in the cases currently at the node.

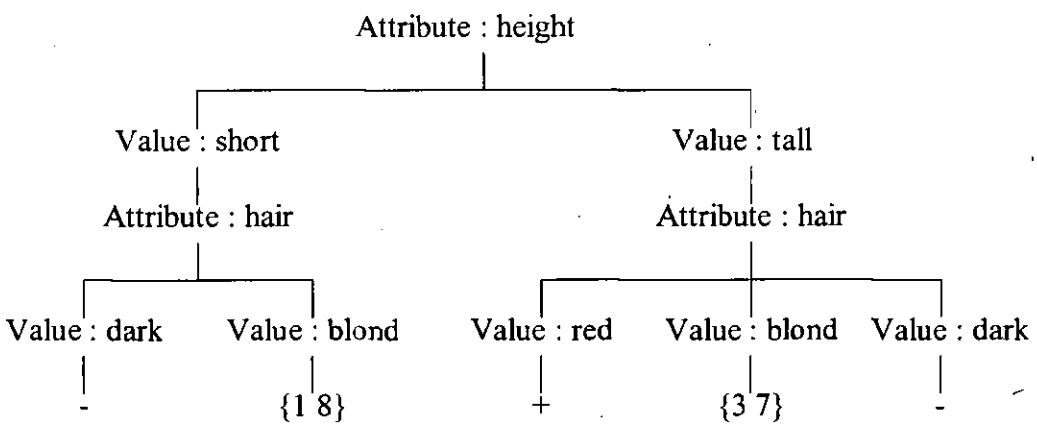


Figure 5.3: The decision tree after the second cycle of the classification algorithm.

For the 'short' branch two legs are created for 'dark' and 'blond' and for the 'tall' branch three legs are created. The input cases available on the 'short' branch are distributed over the newly added legs. As there is one 'dark' case this leg can be labelled '-'. The 'blond' leg has cases 1 and 8 assigned to it and needs further expansion. In a similar fashion the 'tall' branch is expanded to give the tree in figure 5.3.

By applying the algorithm again to unexpanded branches the final tree shown in figure 5.4 is achieved. The tree suggests that anyone who is tall and has red hair is a positive example irrespective of eye colour. Anybody who is tall and has blond hair and blue eyes is also a

positive example. A short person with dark hair is a negative example.

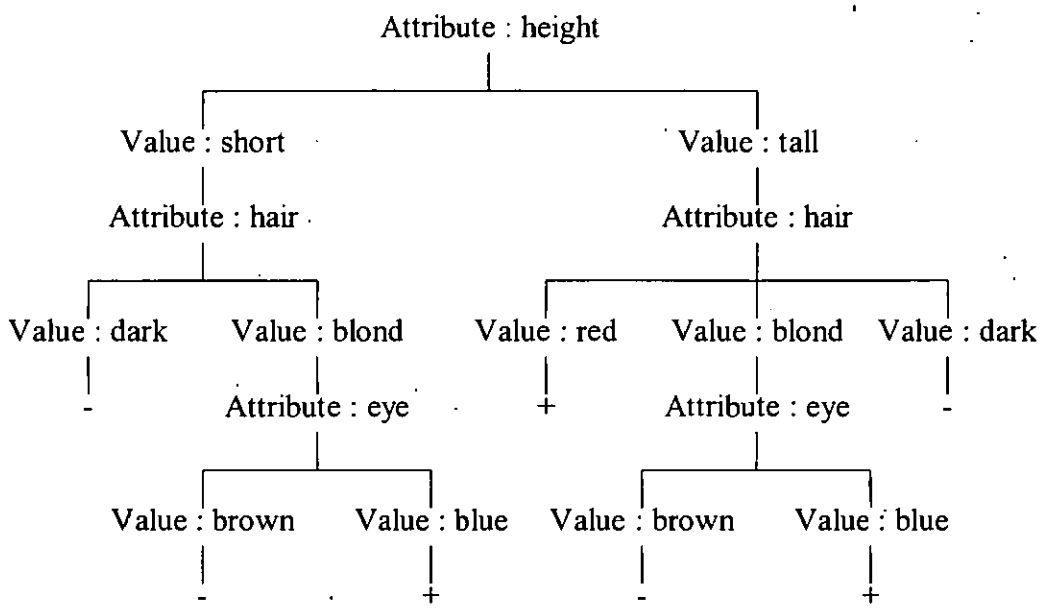


Figure 5.4: The completed classification algorithm decision tree.

Positive and negative classifications can be summarised as two rules; one for positive and one for negative. When forming these rules the levels of the tree are linked with 'and' and the breadth with 'or'.

The negative rule extracted from this tree is:

IF (height is short AND hair is dark)
OR (height is tall AND hair is blond AND eyes are brown)
OR (height is short AND hair is blond AND eyes are brown)
OR (height is tall AND hair is dark)
THEN – NO

The positive rule extracted from the tree is:

IF (height is short AND hair is blond AND eyes are blue)
OR (height is tall AND hair is blond AND eyes are blue)
OR (height is tall AND hair is red)
THEN - YES

The classification algorithm randomly selects the order that attributes are added to the decision tree. In the example above the chosen order was 'height', 'hair' then 'eye' and the resulting tree has seven leaf nodes and is up to three levels deep.

By selecting attributes in a different order, different trees can be produced and some of these trees could be shallower than others. Shallower trees are ones in which the classification is reached in fewer levels. These trees are said to be more efficient as the classification is reached quicker. This point is addressed by Quinlan's ID3 algorithm (Quinlan 1983), which is an enhancement of the classification algorithm, described in the following section.

5.2.3 The ID3 algorithm

The *ID3 algorithm* (Quinlan 1983) is an enhancement to the *classification algorithm* which similarly produces a hypothesis in the form of a *decision tree*; however, the ID3 algorithm adds two new features to the basic *classification algorithm*. These are *windowing* and the *information theoretic heuristic*.

Windowing can be used if the training set is very large. A subset of the training set called the window is chosen randomly to build an initial tree. The remaining input cases are then classified using the tree. If the tree gives correct classification for these input cases then it is accepted for the entire training set and the process ends. If this is not the case then a selection of incorrectly classified instances are appended to the window and the process continues until the tree gives correct classification for the whole set.

Empirical evidence suggests that a correct decision tree is obtained more quickly by the windowing method than by creating a tree from the entire training set. It has been suggested by Thornton (1992), however, that the advantages of windowing are negligible and O'Keefe (1983) has indicated that the windowing method does not always guarantee to find a correct decision tree unless the window uses the entire training set.

The *information theoretic heuristic* is used to produce shallower trees by deciding the order in which to select attributes. The first stage in applying the information theoretic heuristic is to calculate the proportions of positive and negative training cases that are currently available at a node. In the case of the root node, this is all the cases in the training set. A value known as the *information needed for the node* is calculated using the following formula where p is the number of positive cases and n is the number of negative cases at the node.

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{n+p}\log_2\frac{n}{n+p}$$

All available attributes at the node are next considered. Available attributes are those that have not been used so far in the path from the root to the current node and so at the root all the attributes are available.

For each of the available attributes in turn, for each value of the attribute, the number of the cases at the node which are positive and negative are counted. The formula above is used with these proportions to give a value called the *information needed for the attribute*. A value is calculated for every value of the attribute. For example if the 'height' attribute is being considered at the root node a value will be calculated for each of the attribute values 'short' and 'tall'.

A scaled sum of these values is required; the scaling factors are the proportions of the cases at the node with the particular attribute values. In the case of the 'height' attribute at the root the scaled sum will be.

(proportion of tall cases) x (information needed for the tall attribute) +
(proportion of short cases) x (information needed for the short attribute)

This scaled sum is known as the E-score or *entropy* and is subtracted from the information needed for the node to give an *expected information gain for the attribute.*

After repeating for all the available attributes (hair colour, height and eye colour at the root) *the maximum expected information gain* is selected and this attribute is chosen to add to the decision tree. Tree building now proceeds in the manner described for the classification algorithm.

More formally the *entropy* calculations are given by the following:

Assume a set of training instances, C. If attribute A with values $\{A_1, A_2, \ldots A_v\}$ is used for the root of the decision tree, it will partition C into $\{C_1, C_2, \ldots C_v\}$ where C_i contains those objects in C that have value A_i of A. Let C_i contain p_i objects of class P and n_i of class N. The expected information required for the subtree for C_i is $I(p_i, n_i)$. The expected information required for the tree with A as root is then obtained as the weighted average

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

Where the weight for the ith branch is the proportion of the objects in C that belong to C_i. The *information gained* by branching on A is therefore

$$\text{Gain (A)} = I(p, n) - E(A)$$

The ID3 algorithm is summarised as follows:

Input: A training set

Output: decision trees

1. If all the instances are positive then terminate the process and return the decision tree.
2. If all the instances are negative then terminate the process and return the decision tree
3. Else

   Compute the *information gain* for all attributes, select an attribute with the *maximum information gain*, and create a root node for that attribute.

   Make a branch from the root for every value of the root attribute.

   Assign instances to branches.

   Recursively apply the procedure to each branch.

Using the training set of figure 5.1 and applying the classification algorithm by selecting the attributes using the *information theoretic heuristic*, the decision tree in figure 5.5 is generated. This tree has a maximum of two levels and the 'height' attribute does not appear at all.

Again decision rules from this tree can be extracted which would classify a blond haired person with blue eyes as a positive example of the concept and a dark haired person as a negative example.

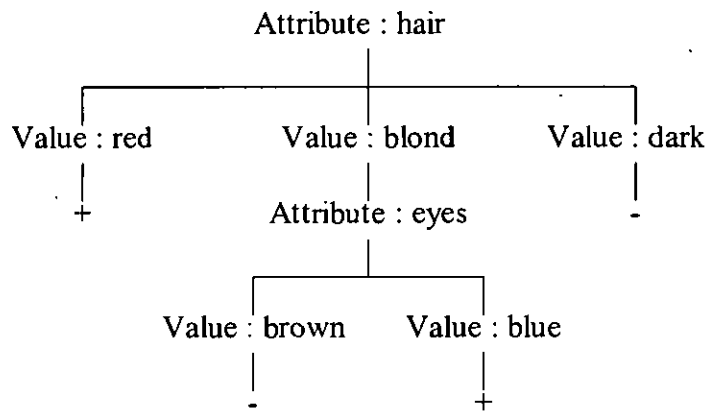


Figure 5.5: The completed ID3 decision tree.

In this example the *information theoretic heuristic* builds the optimal tree and, as a rule, generates compact trees, however, there is no guarantee that the tree will be optimal as the heuristic looks no further than the next attribute to select. This phenomenon is referred to as the *horizon effect*. In this case the eight training instances have resulted in only four leaf nodes and the learned concept is thus more effective at classifying unknown objects.

The next section introduces the C4.5 (Quinlan 1993) algorithm, which is an extension of ID3 that caters for unavailable values, continuous attribute value ranges, and decision tree pruning.

5.2.4 The C4.5 algorithm

C4.5 is an advanced and incremental, version of the ID3 algorithm. The new features of C4.5 are presented below:

Gain ratio criterion: The notion of *maximum information gain* is used in the ID3 algorithm to determine which attribute to select. However, Quinlan points out that if an attribute has a distinct value for each record then this attribute will hold the *maximum information gain* and the training set will be partitioned according to this attribute. Such division in the training data is useless and to avoid this, C4.5 uses the *gain ratio* as follows:

Assume a set of training instances C and attribute A, with values $\{A_1, A_2, \ldots A_v\}$ is used for the root of the decision tree, partitions C into subsets $\{C_1, C_2, \ldots C_v\}$ where C_i contains those objects in C that have value A_i of A.

$$\text{Split Info(A)} = -\sum_{i=1}^{v} \frac{|C_i|}{C} \log_2 \frac{|C_i|}{C}$$

Split Info(A) is the information due to the split of C on the basis of the value of the categorical attribute A.

Gain ratio (A) = Gain(A) / Split Info(A)

With *gain ratio* the algorithm does not only select an attribute containing *maximum information* but also an attribute with minimal partitioning. Quinlan indicates that the *gain ratio* criterion is robust and typically gives a consistently better choice of test attribute.

Unknown attribute values: The C4.5 algorithm can deal with training sets that have records with unknown attribute values by considering the gain ratio for only the records where the attribute value is defined.

Continuous attribute values: C4.5 can deal with continuously valued attributes. For example, if an attribute A has a continuous range and the values are in increasing order $C_1, C_2, C_3, \ldots\ldots, C_m$. Then for each value C_j, j=1,2,3,....,m, the algorithm partitions the records into those that have A_i values up to and including A_j and those that have values greater than A_j. For each of these partitions the gain ratio is calculated and the partition that maximises the gain ratio is selected.

In the next section the application of decision tree building machine learning techniques is presented.

Applications

The application of inductive learning techniques in the field of student modelling has been studied by several researchers (Gilmore and Self 1988). In early ITSs, student models were built from a collection of pre-defined rules and mal-rules (Sleeman 1982) following an extensive protocol analysis of a domain. The feasibility of using inductive learning techniques, to predict a

student's correct and erroneous actions during problem solving have been investigated as an alternative to this.

For example, PIXIE (Sleeman, 1983a) – an algebra tutor – employs decision tree learning to enhance its existing mal-rule catalogue. Similarly, a more recent subtraction-modelling agent called FBM-C4.5 (Chiu and Webb 1998) uses a decision tree learning algorithm to predict the correct and erroneous actions of a student.

In more recent years, the application of machine learning to provide personalised assistance and adaptation, based on patterns in user behaviour, has been investigated.

In the Syskill and Webert (Pazzani *et al* 1997) system a user profile is created based on a rating of the user. A user is asked to classify a number of information pages as either positive examples, in which they interested, or negative examples in which they are not. Using this pre-classified data, the system uses machine learning techniques to generate a new profile, based on the existing one, in order to predict other unseen pages on the same topic. Pazzani *et al* (1997) compared the performance of six different machine learning approaches in their system including ID3 and C4.5 algorithms. They note that the ID3 and C4.5 algorithms were not best suited to their application since they attempted to build trees by testing as few features as possible. There are a number of similarities between the Syskill and Webert system and MLTutor. Both represent pages as Boolean feature vectors; however, MLTutor does not require any pre-classification unlike the Syskill and Webert (Pazzani *et al* (1997)) system.

5.2.2 The focusing algorithm

The focusing algorithm (Young, Plotkin and Linz, 1977) is considered to be a powerful technique to learn concepts. The algorithm aims to produce a definition that is consistent with all given positive training data, but none of the negative. The focusing algorithm uses a *version space* search though the *concept space*. The concept space covers all the possible concept descriptions. The version space only covers concept descriptions which are consistent with the given training instances. The focusing algorithm is similar to Mitchell's Candidate Elimination algorithm (Mitchell *et al* 1986).

In the focusing algorithm the concept being learned is represented by a set of trees. There is one tree for each attribute used to describe the concept.

Suppose the sizes of vehicles used to reach a particular destination are of interest. Attributes in

this example are 'vehicle type' and 'size'. For each of these attributes a tree is required covering all possible instances that can occur. These may be as shown in figure 5.6. These two trees represent the concept space for the example.

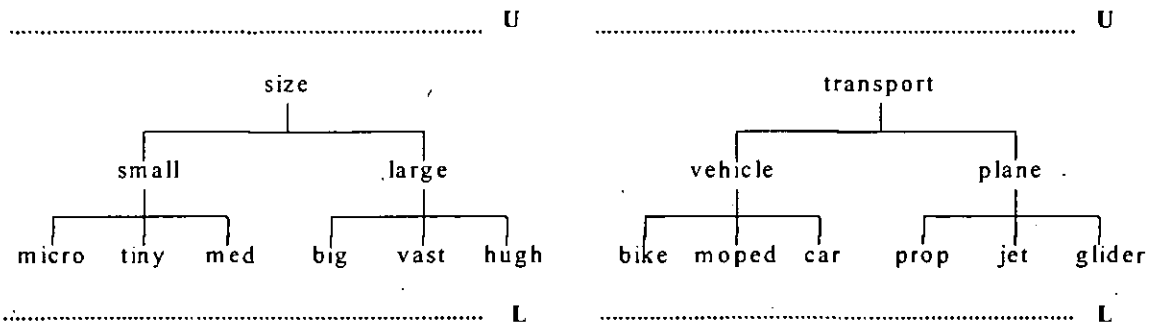


Figure 5.6: The concept space for the vehicle size example.

An upper boundary (U) is initially placed above each tree and a lower boundary (L) is placed below. The version space lies between the U and L boundaries and constitutes the search space for the concept. Concepts are learned my moving the L and U boundaries upward and downwards respectively. Initially the version space covers the entire concept space. The steps of the focussing algorithm are as follows:

For positive instances:	For each tree move L to be coincident with the instance node or to a node above the new and previous instance nodes.
For negative instances:	Select a tree in which U is above the instance-node. Move U towards L to a node above or coincident with L which is not above the instance node. If there is more than one way to do this create a set of version spaces to explore in parallel. This is called a *far miss* otherwise it is called a *near miss*.

The following example applies the focusing algorithm to the training data in figure 5.7 using the concept space described in figure 5.6.

[med moped]	-	positive example of the concept
[tiny car]	-	positive example of the concept
[big jet]	-	negative example of the concept
[med car]	-	positive example of the concept
[med jet]	-	negative example of the concept

Figure 5.7: An example training set given in Thornton (1992).

The first training example [med moped] is a positive example and so the lower bound is moved to be coincident with med and moped as there have been no previous training examples at this stage. The version space after the first training example is shown in figure 5.8.

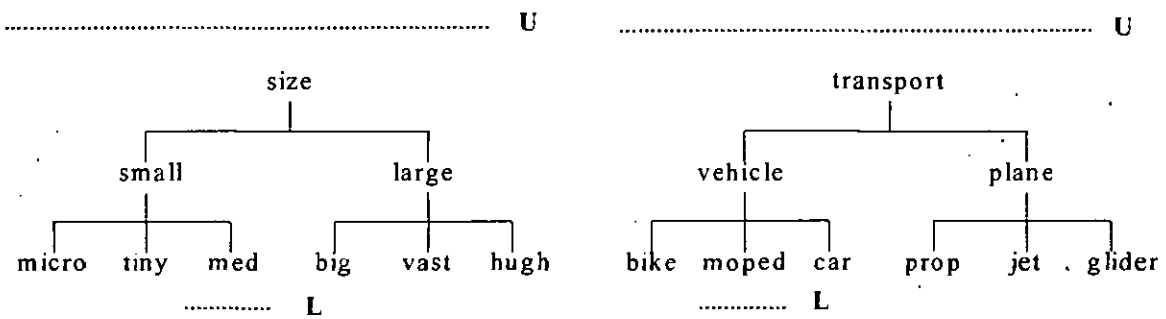


Figure 5.8: The version space after the first training example.

The second training example [tiny car] is again a positive example of the concept. As the modes of transport mentioned in training examples so far are car and moped the lower bound moves to vehicle so as to cover both of these. Similarly the lower bound in the size tree moves to small. The resulting version space is shown in figure 5.9.

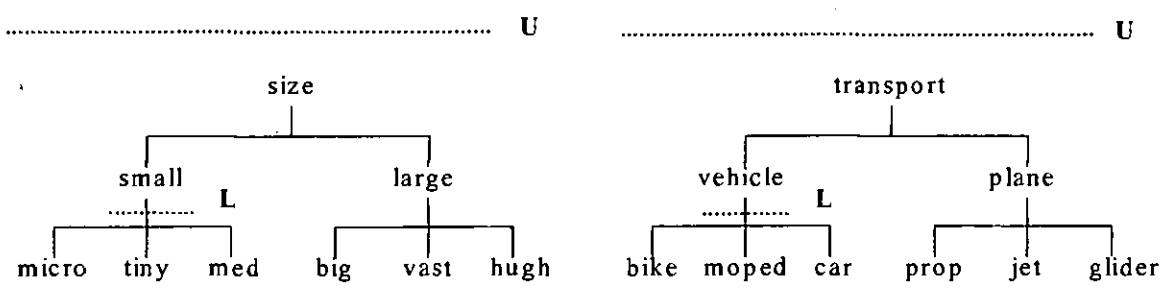


Figure 5.9: The version space after the second training example.

The third training example [big jet] is a negative example. A tree is selected in this case the size tree say and U moved to the small node. It cannot move to the large node as this is above the training instance.

The fourth training example [med car] is again positive but does not result in any changes to the bounds as the lower bound already covers this instance.

The final training example [med jet] is a negative example. A tree is selected in this case the transport tree and the upper bound moved to vehicle. It cannot move to plane as this is directly above the instance. The version space after all the training examples have been processed is shown in figure 5.10.

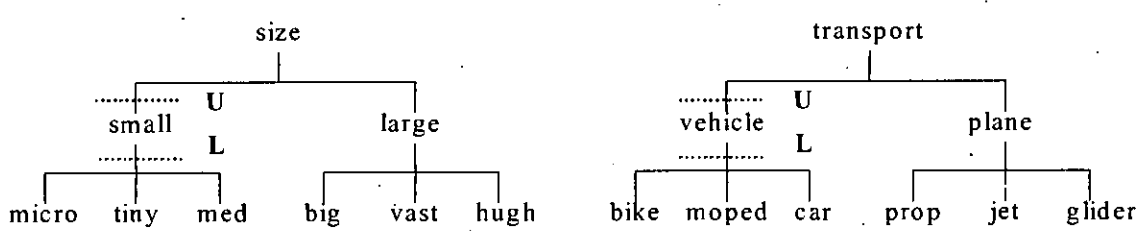


Figure 5.10: The version space after the final training example.

There are no more training examples to be processed. As can seen in both trees the upper boundary U and the lower boundary L have converged at a node. Finally we can describe the concept being learned here as [small vehicle]. This description covers all the positive and none of the negative training data. In this case [small, vehicle] means *small* and *vehicle* and, *small* means *micro* or *tiny* or *medium*. The focusing algorithm cannot handle disjunctive descriptions of a concept and is incapable of representing *tiny* or *medium* and *vehicle*.

Applications

A number of tutoring systems have been developed which attempt to learn concepts by constructing a method of distinguishing members of a concept from non-members. The implementation of concept learning has been considered as the basis for building dynamic student models in ITSs and also for a scheme to allow collaboration between student and tutor.

Self (1987) proposed a system that would use the focusing algorithm to build a type of student model which would function as a *collaborative partner*. The main task of such a partner or internal learner would be to offer advice and suggestions about the teaching material and the learning process. The internal learner would use the focusing algorithm to generate information

about the student by analysing instances examined by the student. Consequently, the tutor could use the information obtained by the internal learner to provide guidance for the student when necessary.

Elsom-Cook's IMPART (1988) tutoring system uses guided discovery methods to teach programming in LISP. It teaches the semantics and syntax of LISP via experiments with an interpreter. IMPART employs a new framework for user modelling called bounded user modelling; it is an alternative to the widely used overlay and perturbation student modelling methods and describes the state of understanding of the learner in terms of upper and lower bounds of the possible states of the learner. The way that the upper and the lower bounds are set is related to focusing.

Although, many attempts have been made to use the focusing algorithm to model human concept learning in ITSs, due to limitations of the algorithm, most have remained theoretical.

Various supervised, concept learning algorithms have been presented. The training data used in these algorithms are labelled positive and negative to indicate their category membership. In the next section, unsupervised concept learning algorithms which due not require pre-classified training data, are discussed.

5.2.3 Clustering algorithms

In general terms, this type of learning does not involve an external teacher providing labelled data for training, and produces classifications based on a measurement of the degree of similarity between objects. Clustering algorithms are usually categorised according to the type of cluster structure they produce e.g. *hierarchical* or *non-hierarchical*, or according to the type of data description language they use e.g. *statistical*, based on numeric descriptors, or *conceptual*, based on symbolic descriptors.

The *non-hierarchical* clustering methods divide a set of N objects into M clusters; no overlap is allowed. These are also known as *partitioning* methods. The *hierarchical clustering* methods produce a nested data structure, called a *dendrogram*, by recursively splitting groups of objects into subcategories until each object is assigned to a subcategory as shown in figure 5.11.

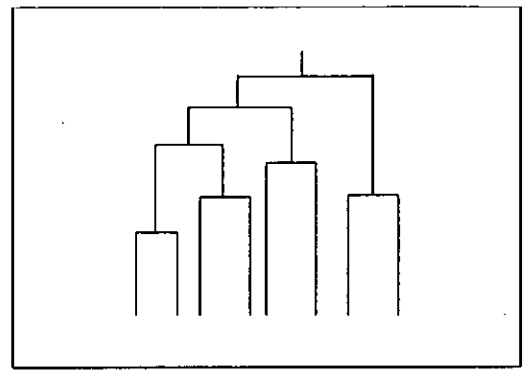


Figure 5.11: Dendrogram of a hierarchical clustering.

The hierarchical methods can be either *agglomerative* or *divisive*. *Agglomerative* clustering places each object in its own cluster and gradually merges these atomic clusters into larger clusters until all objects are in a single cluster. In contrast, *divisive* methods start with all objects in one cluster and subdivide into smaller clusters.

Clustering methods have been employed to identify users' interests (Crabtree *et al* 1998) and to collect evidence about user behaviour in various agent modelling systems (DOPPELGÄNGER (Orwant 1995); Syskill and Webert (Pazzani *et al* 1997)). Researchers have also investigated the potential use of clustering in information retrieval and information filtering systems (Maes 1994; Shetha and Maes 1993).

Stepp and Michalski (1986) note that clustering algorithms based on *numeric taxonomy* fail to take into account background knowledge and also fail to provide meaningful semantic explanations for the resulting categories. That is, they can only produce *extensional* definitions and cannot produce *intentional* definitions of the resulting categories.

Conceptual clustering (Michalski 1980; Stepp and Michalski 1986) addresses these problems. Conceptual clustering algorithms can be used with objects represented by symbolic descriptors and produce simple concept descriptions by applying background knowledge in the formation of categories (Jain and Dubes 1988).

The objective of conceptual clustering is to group objects into conceptually similar classes. Objects to be clustered are described by a number of attributes and the values of these attributes are textual descriptions. In the following example three people are described in terms of their height, weight and occupation.

Attribute	Fred	John	Mary
Height	Tall	Tall	Short
Weight	Heavy	Heavy	Light
Occupation	Manager	Gardener	Doctor

A means of measuring the distance between the object descriptions is required. In statistical clustering methods, the *Euclidean* or *City-block metric* can be used to measure the distance between objects. Other distance and similarity measuring methods, e.g. *Dice coefficient, Jaccard coefficient, Cosine coefficient*, are discussed by Salton (1989).

$$\text{Euclidean Distance} = \sqrt{\sum_i (x_i - y_i)^2}$$

$$\text{City-block Distance} = \sum_i |x_i - y_i|$$

However, these metrics are not appropriate for conceptual clustering methods as values are not numeric. As an alternative, the number of attributes that two objects do not have in common, in other words dissimilarity between objects, or the number of attributes that two objects have in common, in other words similarity between objects are typical metrics used in *conceptual clustering* methods. In the above example using the *dissimilarity metric,* the distance between objects Fred and John is 1 as one attribute value is different and the distance between John and Mary is 3.

The *distance measure* employed has to measure the distances between single objects, but may also have to measure the distance between an object and a cluster (a collection or a group of objects) and between two clusters, depending on the specifics of the algorithm. Various approaches are possible such as the distance to the nearest point within the cluster, the furthest point in the cluster or to a point in some sense representing the centre of a cluster.

A simple clustering algorithm is described by Hutchinson (1994). This algorithm, shown below, produces a partition of the input data in a bottom up manner.

Form a list of all unordered pairs of distinct input points [{x, y} {x, z} {y, z}...]

SORT this list of pairs, into increasing order of distance between the points.

WHILE the list is not empty take the first pair {x,y} from of the list

IF neither x nor y is as yet in a cluster
THEN
form a new cluster C (initially C={x, y})
END-IF

IF one of the two points, say x, is in some cluster C
AND the other point y is not yet in any cluster
THEN
add y to C
END-IF

IF x and y are in different clusters C_1 and C_2
AND x is nearly central in C_1
AND y is nearly central in C_2
AND the distance from x to y is less than the average diameter of C_1 and C_2
THEN
amalgamate C1 and C2 into a new combined cluster
END-IF

OTHERWISE do nothing to the clusters

This algorithm uses the notion of a *cluster diameter*. This is the largest distance between any two points in a cluster. A point is said to be *nearly central* if the distance from it to every other point in the cluster is less than 2/3 of the cluster diameter. The algorithm is said to be non-incremental due to the sort step which requires all data to be available prior to clustering. Should the distance between any two input pairs be the same the location within the sorted list will be arbitrary and could lead to different groupings being produced on the same input.

Applications

ATULA (Milne et al 1996) is an adaptive tutoring system in the domain of Network Theory which employs cluster analysis to identify groups of similar users. In this system, data used in the cluster analysis is obtained from users' test scores on Network Theory, questionnaires and psychological instruments such as the Eysenck personality inventory and the Kolb learning style inventory; academic background, age and gender information about a user are collected by questionnaire; verbal, diagrammatic and numerical capabilities of users is captured by the psychological instruments. The data collected from users is transformed to provide measures of a user's ability in terms of a number of attribute values.

Following cluster analysis on this data, membership lists and *dendrograms* are obtained and used to determine the version of the learning material to be presented. The data collection requirements of the system are extensive and the process is costly in terms of the time spent to build user profiles. It has also been reported by the system authors that using incorrect pre-classified data, due to errors made by students in the initial data collection, had a negative effect on the tutoring outcome.

5.3 Conclusion

A number of machine learning techniques were introduced in this chapter along with a selection of systems which feature a machine learning element.

The MLTutor system is a Web-based information system and contains a machine learning component that is used to analyse browsing behaviour in Web documents. There are a number of similarities between MLTutor and the Syskill and Webert (Pazzani *et al* 1997) system. Both systems represent pages as boolean features vectors; however, MLTutor does not require any pre-classification unlike the Syskill and Webert system. This is achieved in MLTutor by employing a combination of clustering and rule induction algorithms.

Hutchinson's simple conceptual clustering algorithm forms the basis of the clustering within MLTutor. The clustering is used to classify Web pages accessed by a user with a view to identifying commonality in the browsed pages.

The ID3 algorithm, without *windowing*, was used as the basis for the rule induction phase in the initial prototype of MLTutor. The algorithm is used to generate classification rules, which are then used to provide adaptive navigational support. However, the ID3 algorithm was found not

to be ideally suited to this application.

The idea of replacing ID3 with C4.5 in MLTutor was considered, however, the additional features introduced by C4.5 were not seen to offer significant benefit due to the nature of the data in MLTutor. Consequently, the SG-1 algorithm, an enhancement of ID3 was developed. Full details of the SG-1 algorithm and the problems encountered with ID3 are presented in Chapter 7.

Chapter 6

MLTutor Overview

6.1 Introduction

The MLTutor is an adaptive Web-based information system. It has been developed with the intention of testing the feasibility of using machine learning techniques in order to model users' interests from their browsing behaviour.

The MLTutor aims to provide adaptation without the need for any prior knowledge regarding a user's background or interests. The MLTutor uses a novel combination of machine learning techniques to achieve this and this chapter provides an overview of the principals underlying the MLTutor system design. Details regarding the strategy used to evaluate the MLTutor and results of this evaluation are presented in later chapters.

6.2 The MLTutor system design

The MLTutor is a Web-based client server system. The client component of the system incorporates the user interface and runs in a WWW browser. The client captures data which is transmitted to the server using Internet technology. The server component of the system is executed when requested by the client. The server contains a machine learning component which is used to analyse the received data and transmits results to the client. A high level overview of the MLTutor is shown figure 6.1 below.

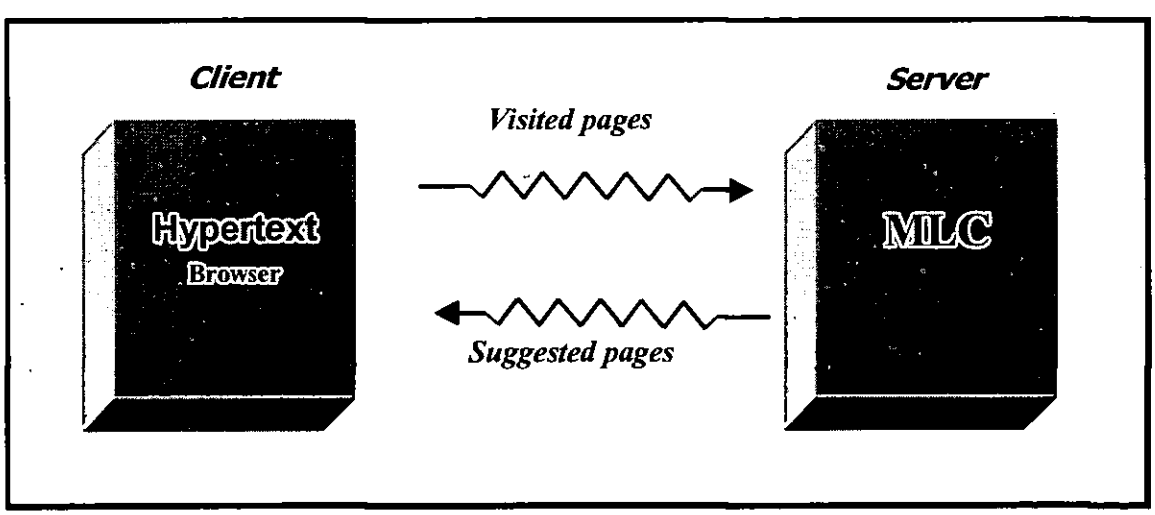


Figure 6.1: An overview of the MLTutor architecture.

6.2.1 Overview of the MLTutor server component

The server based machine learning component (MLC) of MLTutor has been designed so that learning does not require any initial training. This is achieved by the novel integration of attribute-based clustering and inductive learning techniques. Attribute-based learning was discussed in Chapter 5. To facilitate this learning an attribute database is embedded within the system; entries in the database describe pages in terms of presence or not of keywords within the hypertext content of the system. Within MLTutor the catalogue of keywords is based largely on hypertext anchors. The phases of the learning process in the MLC are illustrated in figure 6.2 along with their relation to the attribute database.

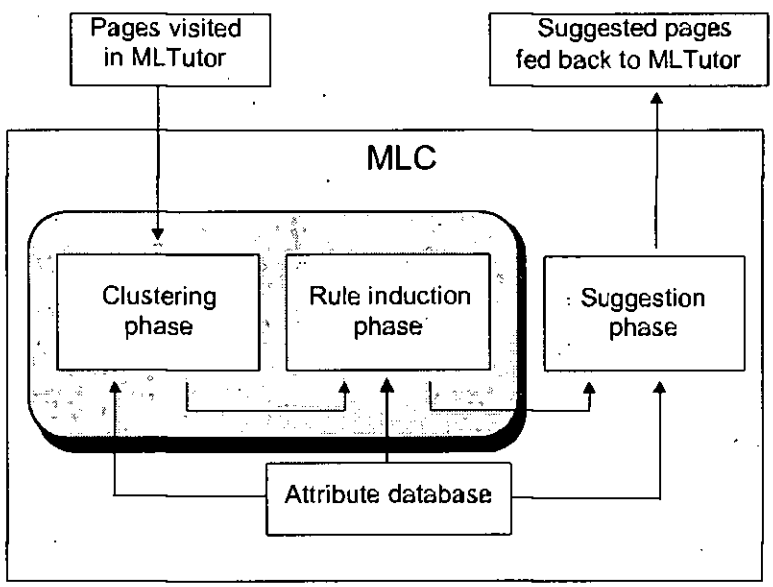


Figure 6.2: The design of machine learning component (MLC).

6.2.2 The application of machine learning within MLTutor

The first phase of the learning process is the clustering phase. As indicated in Chapter 5, clustering algorithms do not require pre-classified training data. In the MLC clustering is utilised to find inherent patterns within the hypertext pages browsed by a user.

The role of clustering in the MLC is illustrated in figure 6.3 below. Let the dark coloured circles in the first ellipse represent the hypertext pages visited by a user out of the available pages, then the second ellipse illustrates the results of clustering these hypertext pages. In this case the clustering has identified three distinct trends within the input and formed three clusters.

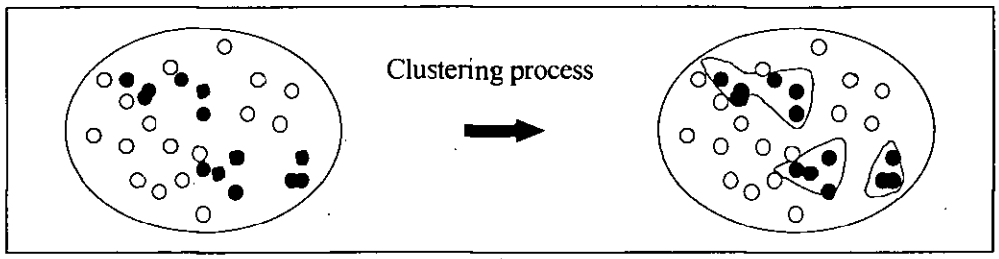


Figure 6.3: Input pages are partitioned into three clusters.

As a result of the clustering, the components of a cluster are in some way related to each other and as such are representative of a concept. Based on this the content of a cluster is assumed to represent positive examples of a concept and anything beyond the boundary of the cluster is taken to be non-representative of that concept.

In the second phase of learning, a rule induction procedure is employed to generate rules which describe the concept of cluster membership. These rules are used to search for other hypertext pages within the original population of pages classified by the rules as belonging to the concept. In figure 6.4 the grey zones extend the cluster boundaries and represent the hypertext pages suggested by the MLC as members of the same concept.

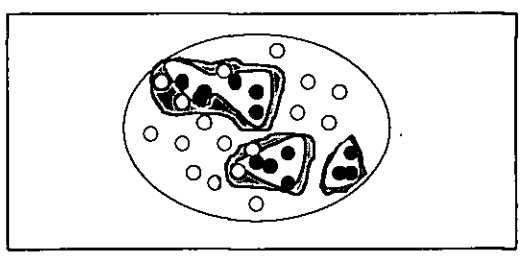


Figure 6.4: The rule induction process allows related pages to be suggested.

The results of the learning process in the MLC are the pages classified as belonging to one of the learned concepts. These hypertext pages are related to those which have already been explored by a user and form the content of the *suggestion list* displayed within MLTutor. As already noted in this thesis a *suggestion list* is a flexible way of introducing navigational adaptation into hypertext.

Applying this novel technique not only eliminates the need for users to provide *relevance feedback*, it also eliminates the need for pre-defined stereotypical user profiles. In most adaptive tutoring environments, real user features are mapped onto one of the system's pre-coded

stereotypes by classification rules - this process is a key element in the adaptation process in many systems. This process is a very laborious task as the domain model of most prototypes is restricted to one domain (Edwards *et al* 1997). Consequently the knowledge base of these systems is likely to be "hand-coded" along with their user modelling component and stereotypes which, as noted by Hohl *et al* (1996), are domain dependent.

In summary, the machine learning approach proposed here facilitates a flexible, individualised approach to adaptation without the need for additional input from a user or pre-classification of users. This is achieved in the MLC by dynamically generating rules which hold generalised information about a user's current area of interest and are used to construct a *suggestion list*. The implementation of the *suggestion list* in MLTutor demonstrates the analysis performed by the MLC; however, the dynamically generated rules, based on the browsing activity, effectively form a profile indicating the users current area of interest at a point in time. This profile is continually updated as further pages are accessed and new rules are created.

6.2.3 Attribute database in MLTutor

The *attribute database* contained within MLTutor is used by the clustering and rule induction processes.

In the clustering process the attribute descriptions of hypertext pages visited by a user are obtained from the *attribute database* and, based on these attribute descriptions, the clustering process puts maximally similar pages into the same clusters.

The rule induction process generates attribute-based rules defining cluster membership. These rules are used to search the *attribute database* for other pages that would be classified as a member of the concept represented by the cluster.

The first stage of attribute database construction is to assemble a list of keywords from the Web pages within the system. In the development of MLTutor a manual process was used to generate a catalogue of keywords which is largely based on embedded hypertext anchors.

Having established a *keyword catalogue* from hyperlinks and other context related words and phrases from the pages contained within MLTutor, a binary vector was created for each page to show presence or absence of each keyword on a page. In the binary representation the absence of a keyword on a page was coded with '0' and the presence with '1'. Each vector forms a single record of the *attribute database*. An example keyword catalogue can be found in Appendix A.2

and attribute database in Appendix A.3.

6.2.4 Potential alternatives to manual coding

The manual approach to coding the *attribute database* is time consuming and would greatly benefit from automation. One option for achieving this would be to index HTML pages within the system using the HTML *keywords* meta tag.

The format for embedding the *keywords* meta tag in a Web page is shown in figure 6.5 and could contain any information. The "*keywords*" information is principally used by Internet search engines to categorise documents.

```
<HEAD>
<META NAME="keywords" CONTENT="life, universe,mankind,
plants, relationships, the meaning of life, science">
</HEAD>
```

Figure 6.5: HTML *keywords* meta tag.

This approach would require consistent indexing across all files within the system to be effective. However, if this is possible it would be a simple task to write a custom program to examine all files within the system to automatically extract keyword information and build the *keyword catalogue* and *attribute database*.

An alternative approach would be to use a tool to generate the *keyword catalogue*. Although outside the immediate scope of this thesis, an alternative approach using techniques from the field of Information Retrieval ought to be feasible. A number of products exist to extract keywords from WWW documents such as DC-dot (UKOLN 1998). DC-dot is a Web-based meta data extracting tool. HTML meta tags, such as keywords, descriptions and other information, including title and headings, are extracted by DC-dot. In order to investigate whether DC-dot could be used for the *keyword catalogue* building process, a number of experiments were carried out. An example can be seen in figure 6.6 below.

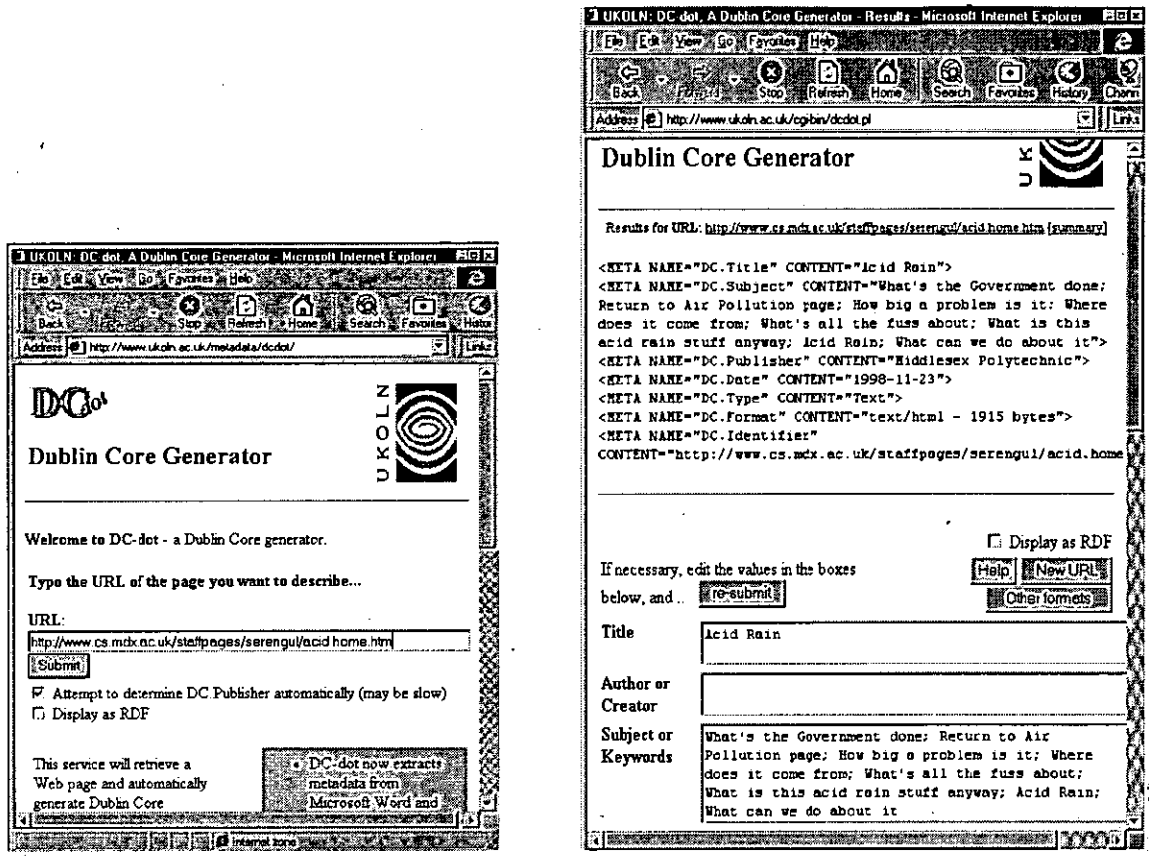


Figure 6.6: Extracted hyperlinks from acid.home.htm file.

In figure 6.6 *meta data* extracted by DC-dot from an example page contains page title and all hyperlinks; the author did not specify keywords on this page. These findings are promising, but overall inadequate for immediate use as the tool is incapable of extracting significant words and phrases which are not hyperlink anchors. These would have to be set up with the keyword meta tag.

In summary, using a tool such as DC-dot with MLTutor would not be straightforward and even the most sophisticated automatic document indexing tools need further manual intervention to get effective results. Although presenting intriguing possibilities for future work, automatic document indexing is outside the scope of this thesis.

6.3 Implementation of MLTutor

The MLTutor prototype has been developed using Web technology. There are two components to the MLTutor; the *client* component encompassing the user interface and the *server* component which incorporates the MLC. The interface of MLTutor has been developed in the

Java language as applets embedded within HTML formatted documents which are downloaded into a standard a Web browser. When the home page of MLTutor is loaded into the browser it partitions the browser into a number of frames as illustrated in figure 6.7. The controlling MLTutor applet is loaded into the left-hand frame and initiates a logon procedure.

Figure 6.7: The MLTutor home page.

Following the *logon* procedure, an index of sites and documents available within MLTutor is loaded into the right-hand MLTutor browser frame. When a Web page is loaded into this frame of the interface, a page applet embedded within the loaded page informs the MLTutor applet which page has been loaded. This is achieved by specifying a parameter to the page applet, which is coded in the HTML source code for the page as shown in figure 6.8 below. The parameter is passed directly to the MLTutor applet when the page is loaded into the browser frame. Although the MLTutor is responsible for loading the initial page into browser frame, the ability to navigate to other pages is controlled by links on the page, and the current implementation of the MLTutor applet cannot directly check which page is currently loaded. For this reason the page applet must inform the MLTutor applet whenever a page is loaded into the browser frame. This mechanism introduces flexibility into the design; if for any reason it is not necessary for the MLTutor to be aware that a page has been visited, the page applet can be omitted and the MLTutor will be unaware that the page has been visited.

```
<APPLET CODE ="C_ReportPage.class" WIDTH=0 HEIGHT=0>
<PARAM NAME ="pageid" VALUE="page006">
<PARAM NAME ="address" VALUE="acid.laws.htm">
</APPLET>
```

Figure 6.8: The HTML code of a page applet.

Data analysis and *suggestion list* generation is performed by the *server*. The processing required to generate the MLTutor *suggestion list* is computationally intensive. For this reason, the MLC of the system was written in C and runs on the *server*.

The MLTutor processes batches of ten pages at a time, the ten page limit was selected to ensure that the learning algorithms are presented with a sufficient volume of data to allow effective learning while at the same time minimising the browsing required before learning can take place.

The MLC described in previous sections executes when requested by the *client*. It takes a measurable amount of time to process the supplied data and generate suggestions. In addition to the processing time there is an associated time for transmission of data to and from the *server*.

Bearing this in mind the MLTutor was designed to automatically re-build the *suggestion list* periodically in the background. This design decision removed the need to devise a mechanism to manually trigger *suggestion list* creation. The automatic time-based generation also prevents the *server* being swamped by requests for *suggestion list* generation if a large number of page are rapidly visited by a system user. To convert MLTutor to rebuild the *suggestion list* following each page selection would require a significant re-design of the system with large portions having to be re-written.

Once the server based MLC has processed the data and generated a list of suggestions, the suggestions are transmitted back to the *client* which displays the suggestions in the suggestion window as shown in figure 6.9.

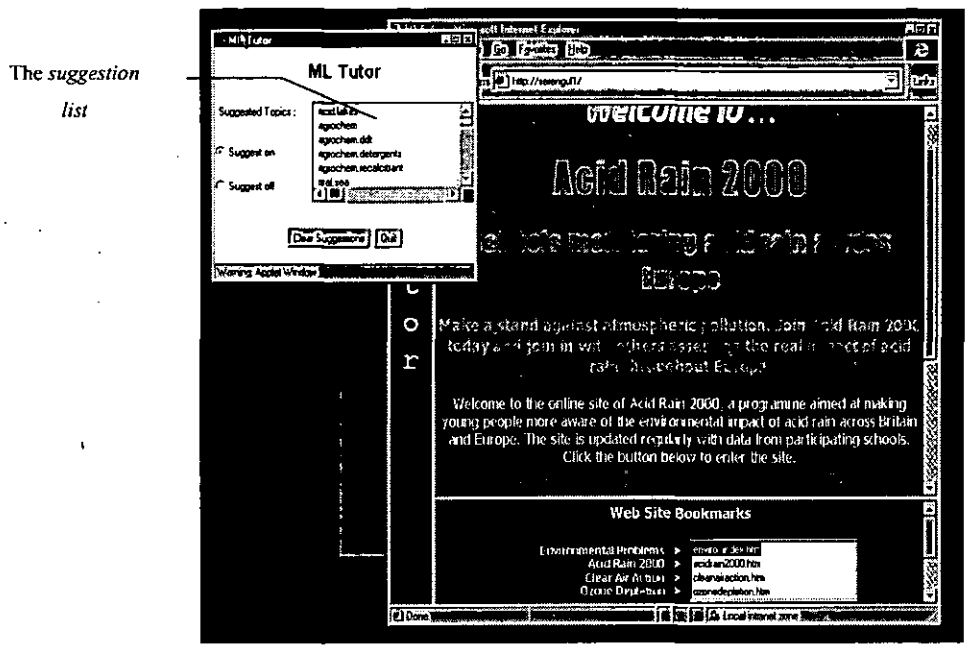


Figure 6.9: The suggestion list pop-up window contains a list of relevant pages.

6.4 Conclusion

This chapter has provided an overview of the principles underlying the MLTutor system design. MLTutor is an adaptive tool that supports Web-based learning. It uses a combination of machine learning techniques to analyse a user's interaction with the hypertext with a view to understanding the aims and motivations for the browsing activity.

As described in this chapter, this analysis is achieved by employing a novel combination of clustering and inductive machine learning techniques. This approach removes the need for any pre-classification of users or their interests and allows information to be obtained unobtrusively without disturbing the natural flow of user navigation.

The machine learning component of the system makes use of an *attribute database*. The *attribute database* is a core component of the system which the machine learning algorithms use to perform their analysis of the user's browsing behaviour. Although a manual approach was employed to build the *attribute database*, some options for automatic *attribute database* generation were considered and the details of these investigations were included.

In the next chapter the initial implementation of MLTutor is described along with the evaluation performed. This evaluation led to a number of enhancement to the systems and the development

of the SG-1 rule induction algorithm. This algorithm is described along with a refined attribute database generation scheme.

Chapter 7

Initial Implementation and Investigation

7.1 Introduction

The high level design of MLTutor was introduced in Chapter 6. The aim of Chapter 6 was to introduce MLTutor at a conceptual level without dwelling on specific technical issues, which are discussed in this chapter.

Within this chapter an initial implementation of MLTutor is described along with details of experiments conducted with it. The experiments highlighted a number of issues with this implementation and these are investigated.

The specific issues covered in this chapter relate to formation of the *attribute database* and development of the SG-1 algorithm for use in the rule induction phase of the machine learning component of MLTutor and the development of alternative cluster selection strategies.

7.2 Experimental evaluation of the initial MLTutor prototype

Based on the overview proposed in Chapter 6, an initial MLTutor prototype was developed. The components of this initial prototype are described in the following sections. In order to evaluate the system a series of experiments were conducted using six participants. The participants were given access to the system and were asked to complete tasks based on the material contained within the system.

7.2.1 Hypertext content of the initial prototype

A 32-page educational document on air pollution was used in the initial experiment. An expert[1] within this domain was asked to set six tasks based on the content of the document.

7.2.2 Attribute databases in the initial prototype

As described in Chapter 6 the *attribute database* of MLTutor is constructed from a catalogue of keywords extracted from the Web pages available within the system. The primary role of hyperlinks embedded in Web documents is to facilitate navigation. As described in Chapter 3, a hyperlink is a connection between a *source node* and a *destination node*. A hyperlink is tied to a specific point within the source node and this is referred to as the *anchor*. This can be a word,

[1] Lian Scholes, PhD student in Water Pollution Control Urban Pollution Research Centre, School of Health Biological and Environmental Sciences, Middlesex University.

phrase, icon, button or image and in a well designed document there will be a clear indication to users about the content of the destination page.

The research in this thesis is concerned with hypertext links in terms of the semantic affiliation between hypertext pages. To assist with a construction of a *keyword catalogue* the following categorisation scheme was devised.

7.2.2.1 Keyword classification

The *keyword catalogue* used by MLTutor is built from two sources of information. These are hypertext anchors built into the HTML document by the author and other content-related keywords and phrases which could legitimately act as hypertext anchors.

Built-in hyperlinks have been split into the following three categories:

Content-related anchors: The HTML language permits web authors to create a direct link between contextually relevant pages of a web document by turning a relevant word or a group of words into HTML anchors. An example of this is illustrated in figure 7.1 below.

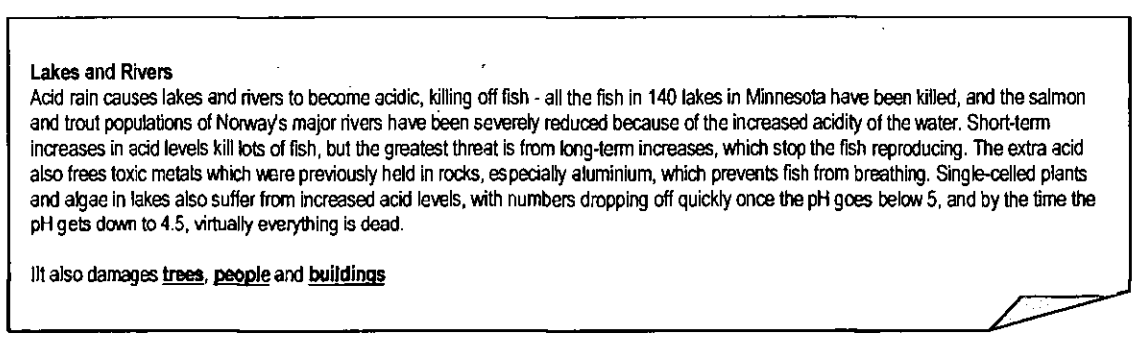


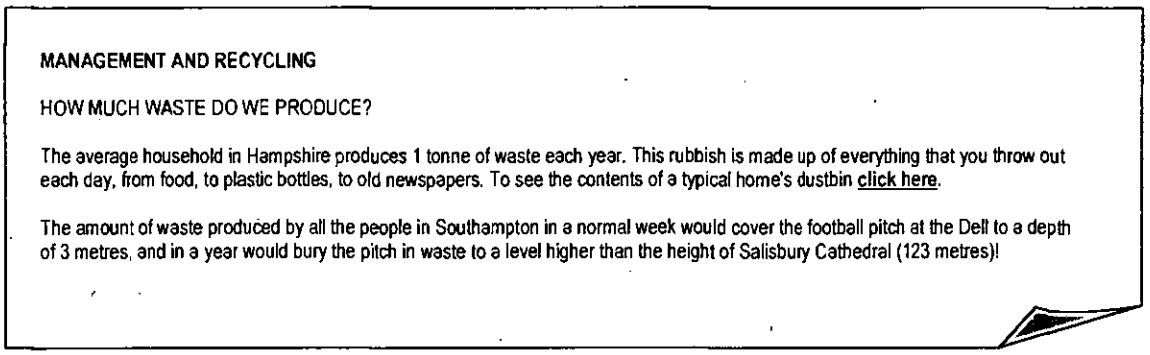
Figure 7.1: A page fragment showing *content-related* anchors

In the page fragment shown above the anchors named 'trees', 'people' and 'buildings' are not the only words pertinent to the topic covered on this page. However, they are directly associated with the Web pages titled 'Trees', 'People' and 'Buildings' and have been selected as anchors by the author of the page. In this study hyperlinks of this sort are referred to as *content-related* anchors.

Auxiliary anchors: Hypertext authors often implement links which are designed to specifically assist navigation. In the context of this study, these are referred to as *auxiliary* anchors.

The page fragment in figure 7.2 displays the application of an *auxiliary* anchor. The anchor

'click here' is a descriptive label rather than a keyphrase and it does not convey any significant value in itself in terms of page content.


MANAGEMENT AND RECYCLING

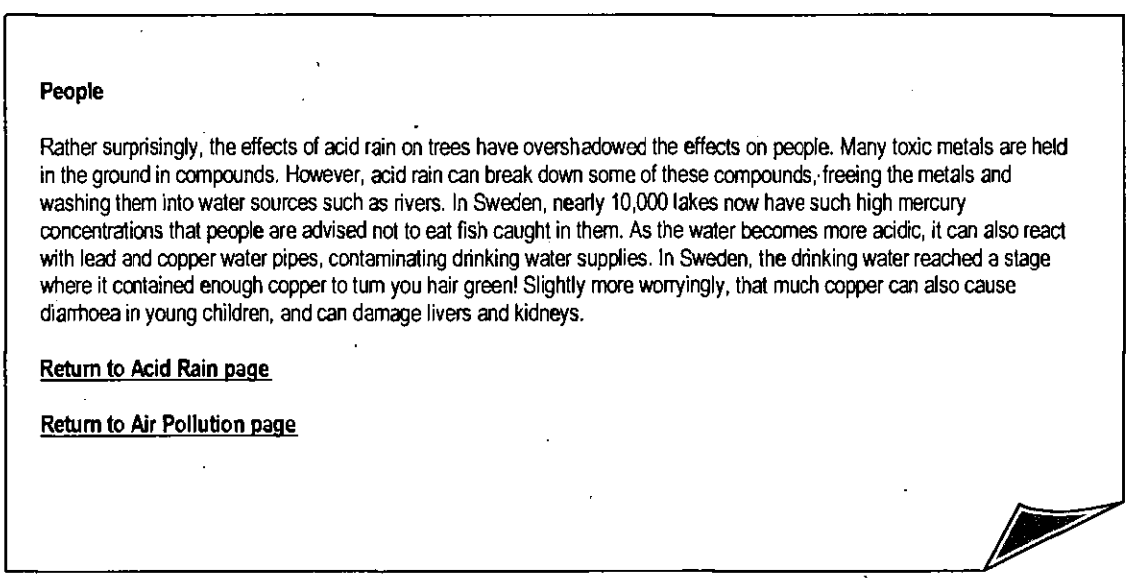
HOW MUCH WASTE DO WE PRODUCE?

The average household in Hampshire produces 1 tonne of waste each year. This rubbish is made up of everything that you throw out each day, from food, to plastic bottles, to old newspapers. To see the contents of a typical home's dustbin click here.

The amount of waste produced by all the people in Southampton in a normal week would cover the football pitch at the Dell to a depth of 3 metres, and in a year would bury the pitch in waste to a level higher than the height of Salisbury Cathedral (123 metres)!

Figure 7 2: The anchor 'click here' is an *auxiliary* anchor.

Generally, *auxiliary* anchors aim to allow users to navigate throughout a document and typically include links to a table of contents, a glossary or an index. The page segment in figure 7.3 contains such features, for example the 'Return to Acid Rain Page' and 'Return to Air Pollution page' anchors link to a table of contents for acid rain and air pollution respectively


People

Rather surprisingly, the effects of acid rain on trees have overshadowed the effects on people. Many toxic metals are held in the ground in compounds. However, acid rain can break down some of these compounds, freeing the metals and washing them into water sources such as rivers. In Sweden, nearly 10,000 lakes now have such high mercury concentrations that people are advised not to eat fish caught in them. As the water becomes more acidic, it can also react with lead and copper water pipes, contaminating drinking water supplies. In Sweden, the drinking water reached a stage where it contained enough copper to turn you hair green! Slightly more worryingly, that much copper can also cause diarrhoea in young children, and can damage livers and kidneys.

Return to Acid Rain page

Return to Air Pollution page

Figure 7.3: A segment from a Web page containing *auxiliary* anchors.

External anchors: This category covers any built-in hyperlinks that lead to other Web sites or e-mail links. In figure 7.4 'Air Quality Reference Guide' and 'click here' are classed as *external* anchors.

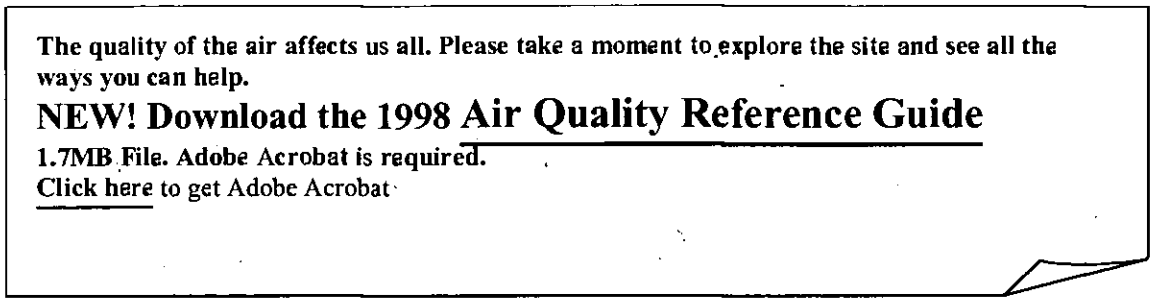


Figure 7.4: A segment from a web page containing *external* anchors.

In a hypertext document not every significant word or phrase will be set as an anchor, but within the context of the page could legitimately have been an anchor if the author had wanted.

Potential anchors: These are keywords which have strong semantic attachment to other topics and could legitimately have been used as hyperlink anchors. The page fragment illustrated in figure 7.5 contains a number of *content-related* keywords such as 'water' and 'acidic' which are *potential* anchors.

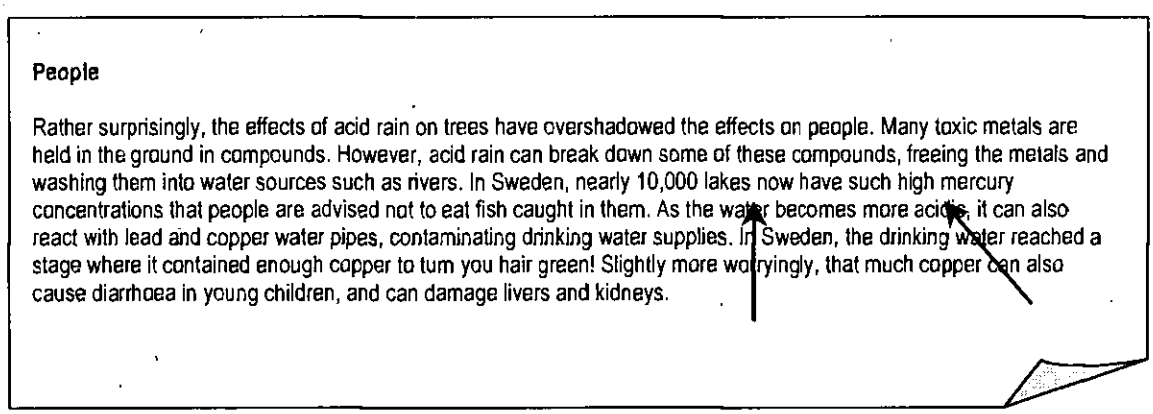


Figure 7.5: A page fragment containing expert recommended keywords.

7.2.2.2 Attribute database construction

Attribute based machine learning techniques are used in the machine learning component of MLTutor as indicated in Chapter 6, consequently *attribute database* construction is a fundamental concern.

In order to investigate the influence of the attribute data on system performance two *keyword catalogues* were created for use in the initial MLTutor prototype; the first catalogue contained 100 *content-related* and *auxiliary* anchors and the second catalogue contained, in addition to

these 100 anchors, a further 29 *potential* anchors recommended by an expert[1] following an examination of the document.

Using these *keyword catalogues*, two separate *attribute databases* were built each containing 32 binary vectors, one for each page of the hypertext document. During the experiments conducted with the prototype, three participants used MLTutor with the 100-*attribute database* and three with the 129-*attribute database*.

7.2.3 Clustering phase of the initial prototype

In the initial construction of the machine learning component of MLTutor, the conceptual clustering algorithm (Hutchinson 1994), described in Chapter 5, was implemented.

7.2.4 Rule induction phase of the initial prototype

In the initial construction of the machine learning component of MLTutor, the inductive learning algorithm ID3 (Quinlan 1986), described in Chapter 5, was implemented without the windowing feature. In order to test the implementation of this algorithm, a standalone test was conducted using Quinlan's original test data. The results of this test, which match those of Quinlan, can be found in Appendix E.9.

7.2.5 Evaluation

The participants who took part in the initial evaluation were asked to use MLTutor to complete a number of tasks. During the experiment the pages visited by the participants and the suggestions generated by the system were recorded.

The navigational paths taken by the participants were assessed by an expert[2] in the field of environmental science. The expert was familiarised with the document contained in MLTutor but was not aware of the tasks the participants were set.

Based on the navigational paths taken by the participants, the expert was asked to recommend related pages, effectively playing the role of MLTutor. Pages suggested by the expert were compared with pages suggested by the MLTutor in order to measure the usefulness of the system-generated suggestions.

[1] Dr Huw Jones, Urban Pollution Research Centre, School of Health Biological and Environmental Sciences, Middlesex University.

[2] Dr Ian Williams, Academic Group Sciences, School of Health Biological and Environmental Sciences, Middlesex University.

The results of the comparison between the expert's suggestions and those of MLTutor were disappointing. Although it was anticipated that some fine-tuning to the *attribute database* would be required – hence the two versions of the databases used in the evaluation – the discrepancy between the system and the expert suggestions were far greater than expected. Typically, while the system suggestions covered all of those made by the expert, additional suggestions were also made resulting in most of the pages within the document being suggested in some cases.

In response to the issues raised by this initial evaluation, a detailed technical investigation of the machine learning component was undertaken.

7.3 Investigation

An assessment of the initial evaluation results highlighted the need for an in-depth investigation into the machine learning component of the system. Two specific areas were identified: *Attribute database* formation and rule induction strategies. These two issues are explored in the following sections.

7.3.1 Attribute database formation

Although two *attribute databases* were used in the initial evaluation, the results of the evaluation failed to provide evidence that one database was more suitable than the other. The fundamental question thus remains: "Do different *attribute databases* have an impact on the machine learning component?".

In order to answer the above question, the conceptual clustering algorithm was applied to the full 32-page document using the two *attribute databases* described previously. The results of the clustering can be seen in table 7.1; the left-hand column of the table shows the clusters formed on the basis of 100 keywords and the right-hand column shows the clusters formed on the basis of 129 keywords.

The pages contained in each cluster are given along with all the attribute values that are contained within these pages. Attributes that were common to all pages within the cluster are indicated in bold and referred to as *primary determinants* of cluster formation. Other attributes are shown in italic and are referred to as *secondary determinants* of cluster formation.

Primary determinants are a measure of similarity between pages and the *secondary determinants* are a measure of dissimilarity. As a measure of cluster quality the ratio of primary

to *secondary determinants* was considered; a higher ratio indicating better cluster quality.

Based on this measure a higher average cluster quality was achieved using 129 attributes, as shown in table 7.1, suggesting that expert recommended *potential* anchors should be retained within the *keyword catalogue*.

Attribute data base containing 100 keyword descriptions	Determinant ratio	Attribute data base containing 129 keyword descriptions	Determinant ratio
Cluster 1 contains: page 1-5-7-8-11 ***common attributes*** **: 1 2 3 4 5 7 8 9 10** ***22***	0.9	**Cluster 1 contains: page 1-5-7-8-11** ***common attributes*** **: 1 2 3 4 5 7 8 9 10** ***22***	0.9
Cluster 2 contains: page 26-27 ***common attributes*****: 4 9 48 49 55 56 57 67 68 69**	1.0	**Cluster 2 contains: page 26-27** ***common attributes*****: 4 9 48 49 55 56 57 67 68 69**	1.0
Cluster 3 contains: page 2-4-6-9-10-12-15-16-17-18-19 ***common attributes*****: 4** ***7 8 9*** **10** ***27***	0.333	**Cluster 3 contains: page 2-3-4-6-9-10-12-13-14-29-30-31** ***common attributes*****: 4** ***7 8 9*** **10** ***27***	0.333
Cluster 4 contains: page 3-13-14-20-21-22-23-24-25-28-29-*30-31-32* ***common attributes: 4 8 9 10 49 55 56 57***	0.125	**Cluster 4 contains page15-16-17-18-19-32** ***common attributes: 4 9 10 27 40 114***	0.166
		Cluster 5 contains: page 20-21-22-23-24-25-28 ***common attributes*****: 4 9** ***49*** **55 56 57**	0.833
Average determinant ratio	0.590	Average determinant ratio	0.646

Table 7.1: Two sets of clusters were created for different attribute settings.

In order to pursue this investigation further attributes 4, 7, 8, 9, 10 and 40, which correspond to *auxiliary* anchors in the classification scheme were discarded from the larger *attribute database* in order to determine their influence on cluster formation. The 32-page document was re-clustered and the resulting clusters can be found in table 7.2.

The *determinant ratio* was again calculated and a higher value achieved compared to the clustering in table 7.1, which suggests that a higher proportion of *content-related* anchors in the *keyword catalogue* results in better clustering. Although *auxiliary* anchors, as described in §7.2.2.1, often do not have a strong contextual relationship with the page that contains them, nonetheless, the page they link to is related to the page containing the anchor. Consequently, in

order to produce the maximum ratio of *content-related* anchors within the *keyword catalogue* all *auxiliary* anchors should be replaced by the page title to which the anchor links.

Attribute data base containing 129 keyword descriptions	Determinant ratio
Cluster 1 contains: page 1-5-7-8-11 ***common attributes*** : **1 2 3 5** *22*	0.8
Cluster 2 contains: page 26-27 ***common attributes*** : **48 49 55 56 57 67 68 69**	1.0
Cluster 3 contains: page 4-18 ***common attributes*** : **20**	1.0
Cluster 4 contains: page 3-2-6-9-10-12-13-14-15-16-17-19-29-30-31-32 ***common attributes***: *27*	0.0
Cluster 5 contains: page 20-21-22-23-24-25-28 ***common attributes***: *49* **55 56 57**	0.75
Average determinant ratio	0.71

Table 7.2: Attribute descriptions 4,7,8,9,10 and 40 are excluded.

Based on the findings of these experiments, a refined strategy for *attribute database* formation was developed.

Attribute database formation principles

- All *content-related* anchors should be included in the *keyword catalogue.* Hyperlinks of this sort should form the bulk of the keywords in the *keyword catalogue.*

- For *auxiliary* anchors which have no contextual meaning within a page, the title of the page to which the anchor points should be used in the *keyword catalogue.*

- All *external* anchors should be excluded from the *keyword catalogue.*
- All expert recommended *potential* anchors should be included in the *keyword catalogue.*

- Having used these rules to build the *keyword catalogue*, the catalogue should be re-evaluated in order to eliminate inconsistencies. The following rules should be applied to the *keyword catalogue.*

- In order to avoid duplicate or even multiple usage of semantically similar words, the synonyms of a word should be removed from the catalogue. For example SO_2 is the standard abbreviation for Sulphur Dioxide and so one term only, say Sulphur Dioxide, should be chosen.

- In situations where a keyword appears on only one page it should be removed from the catalogue.

- All keywords should be converted to lower case for consistency. This ensures that 'Sulphur Dioxide' and 'Sulphur dioxide' are treated equally in *attribute database* construction.

In summary, all *content-related* anchors should be stored in the *keyword catalogue*; for example the anchor 'Acid Rain' should appear in the catalogue as keyword 'acid rain'. For *auxiliary anchors* the destination page title should be stored in the *keyword catalogue*; for example the anchor 'click here' should appear in the catalogue as 'what's in the dustbin?'. All *external* links leading to other Web sites and e-mail links should be excluded from the *keyword catalogue.* Other significant phrases or words, not set as anchors within Web pages contained within the system, but determined by a domain expert as *potential* anchors should be added to the *keyword catalogue.*

7.3.2 Rule induction strategies

The initial evaluation of MLTutor raised some concerns within the rule induction strategy used in the machine learning component. Within the machine learning component of MLTutor the clustering of pages is used to detect the inherent groupings within the hypertext pages browsed by a user, and the decision tree building ID3 algorithm is used to reveal information contained within the clusters.

Typically in decision tree learning instances are represented by a fixed set of attributes. Mitchell (1997) states that decision tree learning is well suited when attributes take on a small number of disjoint values. These map well to the attribute encoding scheme employed by MLTutor which is based on the presence or not of keywords in hypertext pages. In this binary representation '1' is used to indicate the presence of a keyword in the content of a page and '0' to represent

absence. In the literature of decision tree learning algorithms, there is no record of any restrictions on representing attribute values with binary values and a similar scheme has been employed in the Syskill and Webert (Pazzani *et al* 1997) system.

This suggests that a decision tree building algorithm is a suitable candidate for the rule induction phase in the machine learning component. In order to confirm the suitability of ID3 a number of experiments were conducted with this algorithm. For these experiments the *attribute database* with 123 keywords (attributes 4, 7, 8, 9, 10 and 40 excluded from the 129 attribute database as described above) was used to test sets of browsed hypertext pages.

In the following experiment, the MLC was applied to pages 3, 12, 2, 4, 9, 32, 11, 5, 7 and 10. These ten pages are a sequence of pages visited by a participant in the initial evaluation.

Table 7.3 shows the two clusters created for these pages by the conceptual clustering algorithm. As in §7.3.1 *primary determinants* are shown in bold and *secondary determinants* in italic.

Attribute data base containing 123 keyword descriptions
Cluster 1 contains: page 5-7-11 ***common attributes*: 1 2 22**
Cluster 2 contains: page 2-3-4-9-10-2-32 ***common attributes*: *31-84-99***

Table 7.3: Two clusters were created.

In this example the ID3 algorithm was applied to the data in the two clusters to induce decision rules for each cluster. Each cluster in turn was treated as positive training data for the rule induction process; the negative training data being other pages in the cluster not being processed.

The concept descriptions produced by the ID3 algorithm are in the form of *if-then* rules which are used to select other pages to suggest back to the user. The rule developed for cluster 1 of table 7.3 is depicted in figure 7.6.

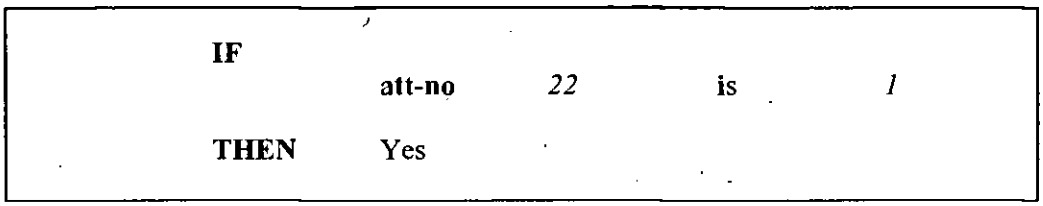


Figure 7.6: The rule generated for cluster 1.

The pages within cluster 1 contain attributes 1, 2 and 22, and the ID3 algorithm needs only a single attribute, in this case 22, to create a rule describing the cluster. Consequently, only pages that contain the keyword represented by this attribute are eligible for suggestion using this rule. Within the context of the MLTutor this is not ideal as the suggestion rule fails to take into account all *primary determinant* of the cluster.

This is due to the fact that ID3 maintains a single current hypothesis as it searches through the space of the given concept. As a result, ID3 is incapable of representing alternative decision trees which are consistent with the available training data (Mitchell 1997).

In order to rectify this weakness the SG-1 algorithm was developed. The SG-1 algorithm is an enhancement to ID3 and full details are presented in the following section.

7.4 The SG-1 rule induction algorithm

SG-1 is a decision tree building algorithm based on ID3. Within ID3, if there are multiple attributes with an equal maximum information gain the algorithm takes into account only one of these. In contrast, SG-1 has the ability to produce multiple decision trees in this scenario and treats each possible attribute equally, building a subtree for each of them. Conceptually, the SG-1 algorithm can be visualised as building three-dimensional trees.

The SG-1 rule induction process is based on the standard ID3 algorithm (Quinlan 1986) with a number of amendments:

- The windowing feature is not implemented.

- An enhancement to cope with situations where a number of attributes could be selected next in the tree building process is included.

The SG-I algorithm is as follows.

Input: A training set

Output: Multiple decision trees

1. If all the instances are positive or negative then terminate the process and return the decision tree.

2. Else

   Compute *information gain* for all attributes. For each attribute with *the maximum information gain* create a root node for that attribute.

   Make a branch from each root for every value of the root attribute.

   Assign instances to branches.

   Recursively apply the procedure to each branch.

The consequence of the ID3 enhancement is described in the following example. Suppose the information theoretic heuristic adds attribute A to the decision tree as the root as shown in figure 7.7. If there are two values for attribute A, two branches are added to the tree and the next attribute is selected.

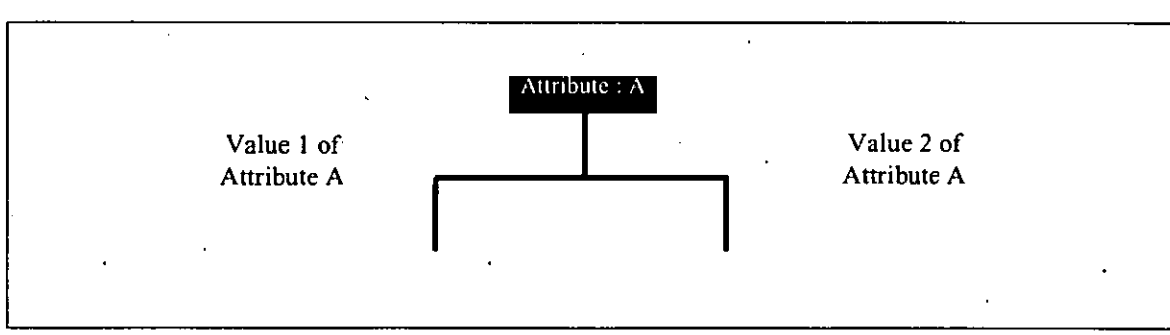


Figure 7.7: Two values for attribute A.

The information theoretic heuristic is used to select the next attribute. If two attributes have the same maximum information gain, say B and C, the ID3 algorithm does not indicate which of these eligible attributes to select. The SG-1 algorithm builds a subtree for each of these attributes. The first subtree as seen in figure 7.8 is built when considering attribute B.

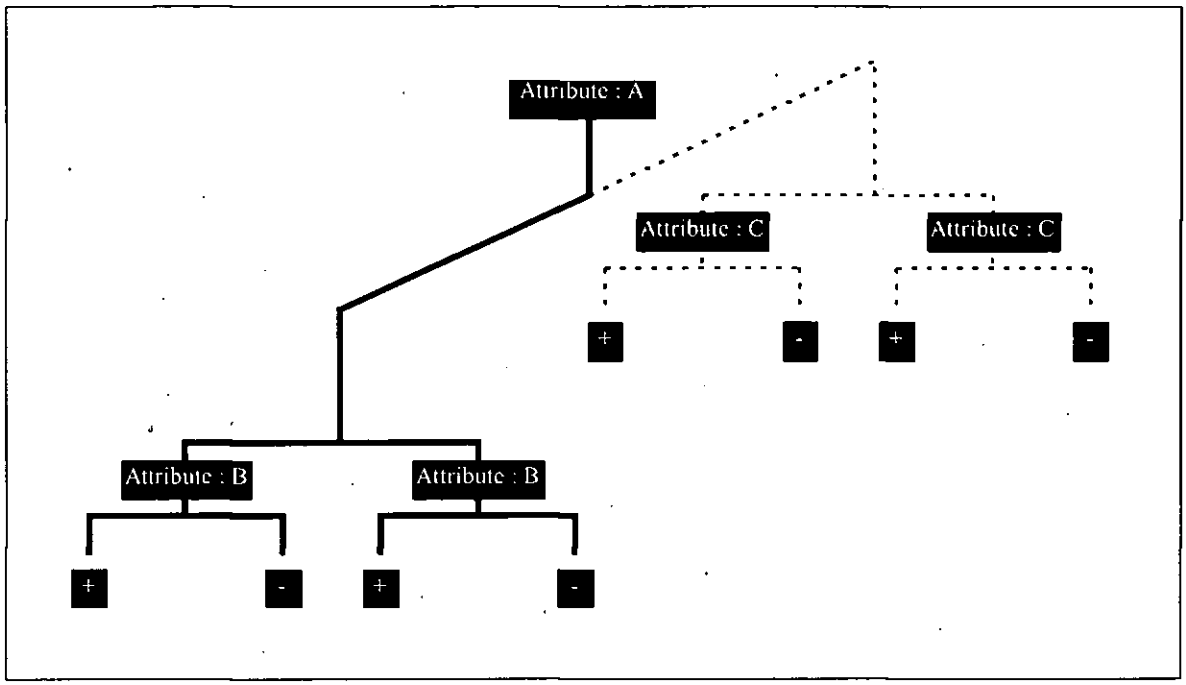


Figure 7. 8: Multiple attribute eligibility, in this case attribute B is added to the tree.

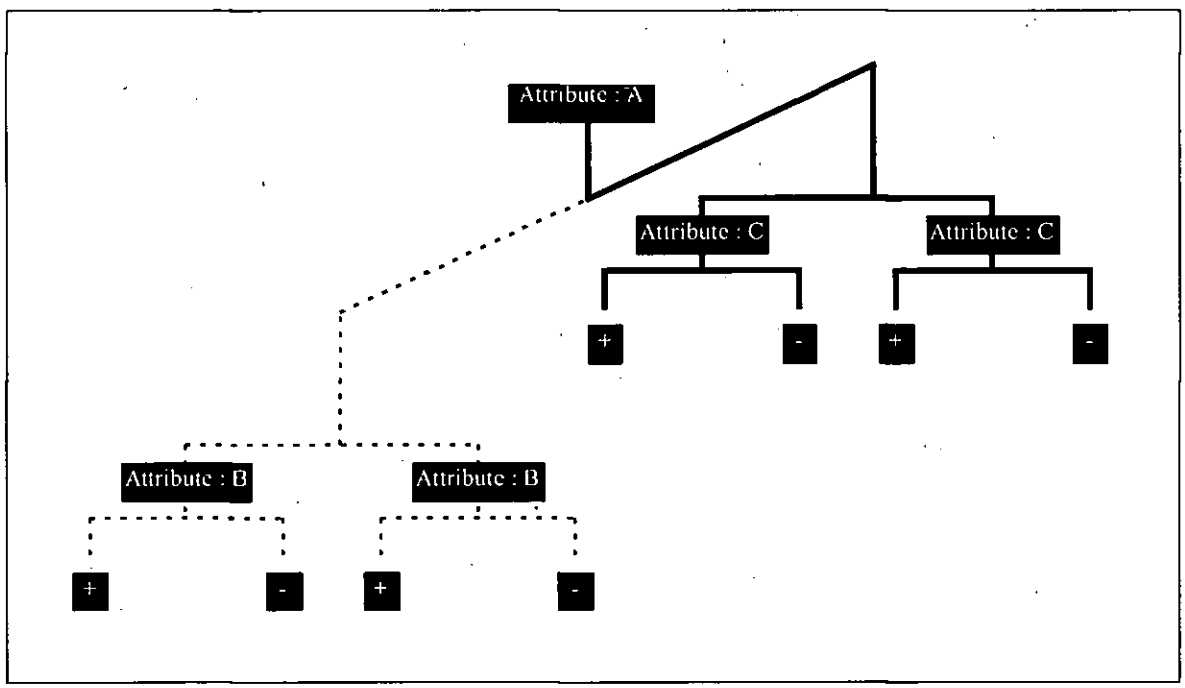


Figure 7. 9: Multiple attribute eligibility, in this case attribute C is considered.

When the SG-1 algorithm considers attribute C, the second eligible attribute, the subtree in figure 7.9 is constructed.

Decision rules from the SG-1 tree are created as previously described for ID3. However, the

additional dimension introduced by SG-1 potentially results in more decision rules being created as SG-1 maintains multiple concept descriptions.

In the examples depicted in figures 7.8 and 7.9, four rules for positive classifications are generated and similarly four rules for negative classifications. In this case the standard ID3 algorithm would have produced two negative and two positive classification rules only. The additional rules produced by SG-1 take into account alternative *primary determinants* from the cluster formation process. By using a disjunction of these rules to search for suggestions, the list of suggestions generated is more representative of the concept the cluster covers.

In order to confirm the impact of the SG-1, the data used to test the implementation of ID3 was amended such that equal maximum information gain is achieved during the tree building process. The results of applying ID3 and SG-1 to this data can be found in appendices E.9 and E.10.

7.5 Cluster selection strategies in MLC

In the initial implementation of MLTutor each cluster produced by the cluster step was processed to generate rules and make suggestions. Based on this strategy the MLTutor suggestions contained most of the pages within the 32-page document.

Given that the test document only contained 32 pages and 10 were required to make suggestions it was not clear if this was a fundamental flaw of the design or a consequence of the document size. In order to investigate further additional experiments were conducted using alternative cluster selection rules.

The results of these experiments were promising and suggestions from the rule induction process were more relevant to recent browsing. This led to the conclusion that a more sophisticated method of cluster selection would be required. However it was decided that a full evaluation could only be conducted with a larger collection of hypertext pages. In order to introduce some diversity within the content and to demonstrate the ability of MLTutor to act as a bridge between independent documents, the collection of pages chosen were from four Web sites with no direct link between them.

In order to pursue the issue of cluster selection further, four versions of MLTutor were constructed for use with a more substantial document. The construction of these versions is

summarised in table 7.4 and full details are provided in the following sections.

	Latest Page	Page Weights
Dynamic clustering	MLTutor Version 1	MLTutor Version 2
Pre-clustering	MLTutor Version 3	MLTutor Version 4

Table 7.4: Summary of the MLTutor versions.

The cluster selection strategies introduced are based on the last page visited in the hypertext and a weighting of the recently visited pages. The last page visited strategy assumes that the last page visited is most relevant to the user, whereas, the weighting scheme takes into account the recent browsing history. In addition to introducing these cluster selection strategies, an alternative to the dynamic clustering used in the initial prototype is introduced. This alternative uses a pre-clustering strategy. In the pre-clustering versions, all pages within the system are pre-clustered and the results stored within the system. The versions that employ pre-clustering do not use the conceptual clustering algorithm dynamically or the SG-1 algorithm to generate suggestions.

7.5.1 MLTutor Version 1

Rule induction based on the cluster containing the latest page visited: Within this version of MLTutor the data passed to the machine learning component of the system is clustered dynamically based on the most recent ten pages visited.

The pages within the cluster that contains the latest page visited by the user are treated as positive training data for the SG-1 rule induction process. The negative training data for the rule induction process are the pages passed to the machine learning component which are not within the cluster containing the latest page visited.

7.5.2 MLTutor Version 2

Rule induction based on the most heavily weighted cluster: In this version of MLTutor the data passed to the machine learning component of the system is assigned a weight based on how recently the page was visited by the user. The most recently visited page is given the highest weight and the least recently visited page is given the lowest weight. A total weight for each cluster generated by the clustering process is calculated by summing the individual weights of the pages that formed the cluster. This strategy aims to reduce the impact of the most recent page being a temporary diversion.

The pages within the heaviest cluster are treated as positive training data for the SG-1 rule induction process. The negative training data for the rule induction process are the other pages passed to the machine learning component which are not within the heaviest cluster.

7.5.3 MLTutor Version 3

Pre-clustering using the latest page: Within this version of MLTutor the complete set of pages available within the system were clustered beforehand and these clusters stored within the system.

The cluster which contains the last page visited by the user is selected and the other pages within this cluster, excluding those in the input data passed to the machine learning component, are selected for suggestion.

7.5.4 MLTutor Version 4

Pre-clustering using weights: Within this version of MLTutor the complete set of pages available within the system were clustered beforehand and these clusters stored within the system.

The data passed to the machine learning component of the system is assigned a weight based on how recently the page was visited by the user. The most recently visited page is given the highest weight and least recently visited page is given the lowest weight. Again, this strategy aims to reduce the impact of the most recent page being a temporary diversion.

A total weight for each cluster is calculated by summing the individual weights of the input pages they contain. The heaviest cluster is selected and the other pages within this cluster, excluding those in the input data passed to the machine learning component, are selected for suggestion.

7.6 Conclusion

This chapter has presented an in-depth technical description of the initial prototype of MLTutor along with details of the evaluation conducted with it.

Evaluation of the initial prototype incorporating the simple conceptual clustering and ID3 algorithms identified a number of weaknesses, which were described in this chapter. An

investigation of these weaknesses led to the development of principles for the formation of the *attribute database* and the development of the SG-1 algorithm.

This investigation also indicated that rules for cluster selection would need further investigation and to further this investigation, the design for four variants of the MLC was presented. These incorporate alternative cluster selection strategies based on last page and weights, two of the variants use a pre-clustering strategy and the other two a dynamic clustering strategy. Variants of the MLTutor using these four MLC versions along with a non-adaptive version of MLTutor were used in a comparative study.

The framework for this comparative study is described in the next chapter and the results of the evaluation presented in Chapter 9.

Chapter 8

Evaluation Framework

8.1 Introduction

The evaluation of adaptive hypertext systems is a very active issue (Fumer *et al* 1999), but there are still no agreed methods for performing this task. In order to highlight current trends in this field, this chapter outlines a number of measures used in the evaluation of adaptive systems and discusses the evaluation results of these systems.

Following this the strategy used in the evaluation of MLTutor is defined. The MLTutor evaluation primarily investigates the feasibility and utility of applying a variety of machine learning techniques to support adaptation within a WWW-based information system. The analyses of the results of the MLTutor evaluation are presented in Chapter 9.

8.2 Evaluating adaptive hypertext systems

Evaluating a system is an intricate task and as noted by Hook (1997) it becomes more difficult if the system is adaptive. In recent years, although adaptive hypertext research has produced promising results (in terms of assisting personalised information gathering in an educational context) as reported by Brusilovsky (1996), a weak point of this research field is the lack of comprehensive empirical studies to measure the usefulness of adaptation within such systems and between such systems. One reason for this is that there is no standard or agreed evaluation framework for measuring the value and the effectiveness of adaptation yielded by adaptive systems.

A typical approach to determining the effectiveness of adaptation has been to compare the performance of an adaptive system against a version of the system with adaptation disabled. Hook (1997) states that adaptivity should preferably be an inherent part of a system, and so if it is removed from the system, the system may not be fully functional.

However, as noted by Brusilovsky and Eklund (1999), users may or may not accept the adaptivity available, which they also report can initially result in a poorer performance. For this reason the research in this thesis is based on the principal that adaptivity should be made available as an optional facility, allowing users who feel confident with the feature to take advantage of it, while at the same time not penalising those users who are not so confident.

In the MLTutor the suggestion list is the adaptive component of the system. The suggestion list has been developed as a *plug-in plug-out* component; with this feature disabled the MLTutor

system is a simple hypertext browsing system.

Nelson (1992) reports that the majority of studies to date have utilised empirical, quantitative data analysis to measure the efficacy of system performance. The findings reported in the following section, which reviews the criteria used to evaluate various systems, support Nelson's claim.

8.2.1 Evaluation criteria

Various criteria have been used to evaluate adaptive systems. A number of these are described in this section, with reference to adaptive systems whose evaluation has used the criteria.

The review of evaluation criteria makes reference to the following systems:

CercleText	(Taylor and Self 1990)
HYPERFLEX	(Kaplan et al 1993)
Metadoc	(Boyle and Encarnacion 1994)
ISIS-Tutor	(Brusilovsky and Pesin 1995)
ELM-ART II	(Weber and Specht 1997)
InterBook	(Brusilovsky and Eklund 1999)

Comprehension and learning

The primary hypothesis of Boyle and Encarnacion (1994) predicted that Metadoc users would have better reading comprehension compared to stretchtext (see §4.4.2) and hypertext users, and so users would be able to give more correct answers to questions. The results from reading comprehension tasks were confirmed by Boyle and Encarnacion to be consistent with this hypothesis.

Weber and Specht (1997) postulated that both visual adaptive annotation and individual curriculum sequencing with a 'NEXT' button would motivate users of the ELM-ART II Lisp programming tutor (discussed in detail in Chapter 4). However, the findings support their hypothesis only partially; the 'NEXT' button had a significant effect when users had no previous programming experience, but no effect on users with experience. Link annotation was reported as having had no effect at all during learning for users without programming experience.

In the evaluation of InterBook (Eklund *et al* 1997), students test scores were used as a criterion

to measure the effects of adaptation. The results of the study indicate that there was no significant statistical difference between the scores gained by users who performed tasks with, and those who performed tasks without adaptive navigational support.

Taylor and Self (1990) compared user performance in CercleText with and without the Topic Map, which is used as a support tool, by checking the accuracy of a user's answers against those of domain experts. However, the findings, as reported by the system developers, indicated that no significant performance differences could be found.

Time spent

The measurement of time spent to complete a task, all tasks or to learn a concept is a frequently used criterion for system evaluation.

In HYPERFLEX, Kaplan *et al* (1993) used the length of time spent by a user on a topic as a means of measuring the user's interest and also reported that HYPERFLEX can reduce time spent searching for information by up to 40%.

Boyle and Encarnacion (1994) based their experimental study of Metadoc on two main hypotheses. The primary hypothesis was based on *comprehension and learning* as described above. The secondary hypothesis predicted that Metadoc users would spend less time, visit fewer nodes and perform fewer operations in order to locate information than the users of hypertext and stretchtext. Boyle and Encarnacion reported that the time spent results appeared to be partially consistent with the secondary hypothesis, in the sense that Metadoc users spent less time than hypertext users, although stretchtext users spent the least time overall. However, these results were reported as not being statistically significant.

Time based measures were also used in the evaluation of the InterBook (Brusilovsky and Eklund 1999), however, these measures did not produce any significant data. The InterBook developers reported that adaptive link annotation did not result in audit trails containing fewer nodes visited, reduced transitions between concepts and more time spent on nodes.

Time spent during learning was also measured for the ISIS-Tutor (Brusilovsky and Pesin 1995) evaluation. The results of the evaluation show that adaptive presentation (see §4.4.1) can reduce time spent during learning and improve comprehension. In contrast, the effect of adaptive link annotation on learning time was reported as minimal.

In CercleText (Taylor and Self 1990), time was recorded to determine when the system's features were activated and deactivated.

Number of nodes visited

Counting the number of nodes visited is another commonly used evaluation criterion. Boyle and Encarnacion (1994) used this criterion in the Metadoc evaluation. They report, however, that an analysis of the number of nodes visited did not confirm their primary or secondary hypotheses, described above.

Node counting was used in the evaluation of ELM-ART II (Weber and Specht 1997). The evaluation results suggest that adaptive annotation does not have any systematic effect on the number of navigational steps taken by users. However, Weber and Specht interpret this finding as a consequence of having a small test group.

Brusilovsky and Eklund (1999) used the number of nodes visited as an indication of a user's interest in a topic. They believe that adaptive annotation would produce a more 'purposeful' learning environment and consequently users would need to visit fewer nodes. As reported above under time based measures this belief was not confirmed.

The number of nodes visited was also used to evaluate ISIS-Tutor (Brusilovsky and Pesin 1995). The reported results indicate that adaptive presentation failed to reduce the number of nodes visited however adaptive annotation did.

Number of nodes re-visited

This criterion has not been used in the evaluation of many adaptive systems. However, in the evaluation of ISIS-Tutor, Brusilovsky and Pesin (1995) consider this factor as a measure to determine transitions either from concept to concept or between index and concepts. They report that the tests produced results in favour of adaptive hypermedia.

Navigational tool usage

Based on data in user audit trails, Brusilovsky and Eklund (1999) noticed that some of the adaptive InterBook users almost never used annotated links (by which adaptation is provided) and some of them used the feature reasonably often. As a result of this the audit trails of users who ignored the adaptivity were excluded from a detailed evaluation. Findings of the evaluation suggest that use of adaptive annotation initially resulted in a significantly poorer performance. Brusilovsky and Eklund suggest that a complex interface design may have distracted users from

the content as most of the navigational choices were made without adaptive annotations. They comment 'user model based link annotation seems to be of value to those that agree with it'.

Paths of users navigation

Taylor and Self (1990) studied user performance in versions of CercleText with and without the Topic Map. They mapped a user's navigational path onto an expert specified 'ideal' path in order to judge performance. This information was used to determine how frequently the users got lost and required guidance.

Background knowledge

The level of background knowledge possessed by the participants in an evaluation can have a bearing on the results of the evaluation.

Boyle and Encarnacion (1994) note that Metadoc has greater impact on novice user compared to expert users. Similarly, Weber and Specht (1997) comment on the effects of prior knowledge on the evaluation of ELM ART II.

This review of the evaluation strategies used in a number of adaptive systems has identified a number of common approaches as described above. In the following section the framework used in the evaluation of MLTutor is described.

8.3 Evaluation of MLTutor

The MLTutor evaluation primarily aimed to investigate the feasibility and usefulness of applying a machine learning approach to modelling users' interests from their browsing behaviour in a WWW-based information system. A secondary aim of the evaluation was to demonstrate the robustness of the machine learning algorithms used in MLTutor. The following focuses on the first of these aims.

8.3.1 MLTutor evaluation strategy

In order to evaluate the effectiveness of the machine learning component of MLTutor an empirical study was conducted. The evaluation aimed to assess the feasibility and also the utility of using machine learning techniques for the analysis of an individual users' navigational pattern by comparing the performance of MLTutor versions containing the four variants of the machine learning component described in Chapter 7. The four variants were compared with

each other and against a non-adaptive control version based on the framework outlined in the following sections.

8.3.1.1 Experimental set-up

User instructions: An instruction sheet illustrating the MLTutor logon procedure and use of the navigational tools available within the system was prepared. A version of the instructions, from which the sections describing the adaptive features were excluded, was also prepared for users of the control version (see Appendix C).

User tasks: An expert[1] in the field of environmental science set 13 questions (see Appendix C.3) based on the information covered in the MLTutor system (see Appendix A.1) and provided model answers (see Appendix C.4) indicating the hypertext pages which contained the answers. Answers to some of these questions specifically required at least two Web sites to be explored by the participants of the experiment. As there was no built-in link between the four available Web sites, a bookmark facility was provided to allow jumps between them.

Participants: Thirty people with differing academic backgrounds participated in the empirical study. The backgrounds of the participants who took part in the experiment are shown in table 8.1 and the details of each individual participant can be found in Appendix D.1.1. Each group within table 8.1 consisted of five participants, who were each asked to complete the set tasks set by the expert using one of the five variants of MLTutor. As noted above one variant of MLTutor had all the adaptation disabled and was effectively a simple hypertext system and the other four adaptive variants had differing formulations of the machine learning component.

Group number	Background	Occupation
Group 1	Environmental Science	3rd year Bsc students
Group 2	Environmental Science	3rd year Bsc students
Group 3	Geography, Engineering	4 academics and 1 professional
Group 4	Environmental Science	PhD students
Group 5	Computing Science	PhD students
Group 6	Computing Science	PhD students

Figure 8.1: MLTutor evaluation participants.

[1] Dr Ian Williams, Academic Group Sciences, School of Health Biological and Environmental Sciences, Middlesex University.

8.3.1.2 Raw data collection

User feedback questionnaire: In order to gather feedback on various aspects of the system a user feedback form was prepared consisting of two sections. In the first section, six questions asked for feedback on a scale, in the second section, the participants were given the opportunity to comment on various features of the system and provide any other feedback. A sample feedback questionnaire form can be found in Appendix C.

Answer Sheets: In the experiment, participants were asked to answer 13 questions while browsing the Web documents within MLTutor and provide their answers in written form. The written answers of the participants were assessed against the model answers.

Log files: During the experiment each participant's interaction with MLTutor was recorded in a log file. The MLTutor log files were used to identify the paths taken through the hypertext to reach pages containing the answers to the set tasks.

Log files are the most effective method of unobtrusively determining how often adaptive features are being used and to trace interaction with the system. Figure 8.2 presents an annotated fragment of a sample log file.

The sample log file fragment illustrates the data recorded during a user's interaction with MLTutor. The features identified in figure 8.2 are described below.

- Visited page name: The name of the page visited was mainly used to identify if the pages containing the model answers have been visited.

- Visited page access time: Access to every page, irrespective of access method, was time stamped.

- Suggestion discarded: This information indicates that the keyword list for a *suggestion list* entry has been looked at, but the suggested page itself has not been visited.

- System suggested pages: These pages have added to the MLTutor suggestion list following an analysis of browsed input by the machine learning component of the system.

- Website transition: This information was not recorded explicitly in the log file but was easily identified by an analysis of page transitions which were recorded in the log file.

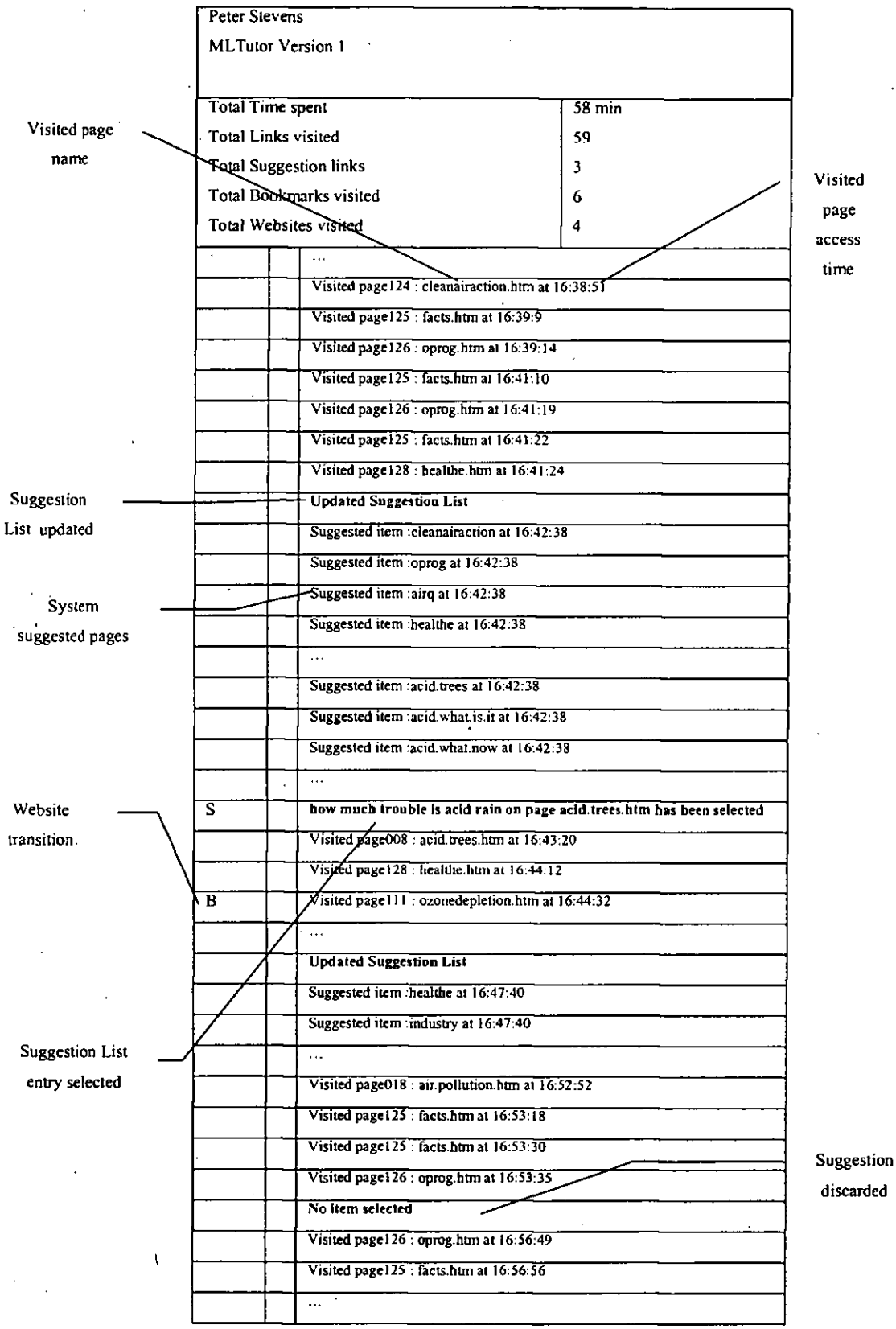


Figure 8.2: A fragment of a sample log file.

- Suggestion list updated: This information indicates when the *suggestion list* was re-generated.

- Suggestion list entry selected: This information indicates that the user accessed the page by following the system's recommendations.

8.3.1.3 Measures

Chapter 9 presents the results of both the empirical analysis and the technical analysis of the MLTutor. The technical analysis aims to assess the robustness of the machine learning techniques employed in MLTutor. The empirical analysis aims to assess the benefit of using adaptive MLTutor compared to the non-adaptive MLTutor.

The empirical analysis was based on data extracted or calculated from the three raw data sources described in §8.3.1.2, collected while following the experimental procedures described in §8.3.1.1. The following data was extracted or calculated for the empirical analysis:

Participant task results: These were used to see if results were significantly better using an adaptive MLTutor version, and if so which one, compared to the non-adaptive version.

Background Knowledge: Participant scores were analysed based on the group classifications in table 8.1 in order to confirm that background knowledge had a bearing on the results.

Time spent to complete tasks: Calculated from log files, this data was used to see if total task completion time was reduced by any MLTutor version.

Total link usage: Extracted from log files, this data was used to see if total link usage was reduced for any MLTutor version, suggesting more purposeful browsing.

Total suggestion list link usage: Extracted from log files, this data was used to see if suggestion list link usage was higher for any MLTutor version, suggesting more focused suggestions.

Total bookmark link usage: Extracted from log files, this data was used to see if bookmark link usage was reduced for adaptive MLTutor versions compared to non-adaptive.

Navigation paths: Data within the log files was analysed to determine how participants reached pages containing answers to the tasks; either directly or via the suggestion list, and whether

tasks were answered correctly from these pages.

8.4 Conclusion

A number of issues relating to the evaluation of adaptive system were presented in this chapter along with the evaluation criteria used by a number of researchers.

The evaluation of adaptive system is a complicated task and no agreed evaluation framework exists. The approach employed by many researchers is based on a comparison of the adaptive system with a non-adaptive version. This approach is only feasible when removal of adaptivity leaves a fully functional system.

Due to the nature of the adaptivity provided by the MLTutor this approach is feasible and the framework for a comparative empirical evaluation of MLTutor was described in this chapter. In part, the criteria used in the evaluation of MLTutor are similar to those used in the evaluation of the other adaptive systems and these criteria were detailed. However, an original approach based on analysing the paths taken to answer tasks was also used to evaluate MLTutor.

An analysis of the data captured during the empirical study is presented in Chapter 9.

Chapter 9

Results

9.1 Introduction

The previous chapter described the framework used to evaluate the MLTutor system. This chapter describes the analysis applied to the data gathered during the experiment and presents the results.

In the first part of this chapter (§9.2) an empirical analysis of the collected data is presented. In the second part of the chapter (§9.3) a technical analysis of the machine learning component of MLTutor is given.

9.2 Empirical evaluation of MLTutor

The objective of the empirical study was to measure the benefit of the adaptive navigational support provided by MLTutor during a goal-oriented information search activity.

As described in the previous chapter (§8.1.3.2) data from three sources was available to meet this objective.

- Participants' answer sheets
- Feedback questionnaires
- Log files

The data from these sources was analysed individually followed by an analysis using data from several of the sources. The results of the analysis are presented in the following section.

9.2.1 Analysis of the participants' answer sheets

In order to proceed with the answer sheet analysis the participants' written answers were assessed against the expert recommended model answers. The participants' scores for each question along with the overall scores can be found in table D.2.1 of Appendix D.

In the first phase of this analysis the participants' scores were evaluated in terms of their background knowledge on environmental science and computer experience. The results for the groups are compared as a whole in table D.2.2 in order to take into account any impact from the various MLTutor variants used by members of the groups.

The results of this analysis are consistent with those of Boyle and Encarnacion (1994); the 4th

test group, with greater computer experience and an assumed deeper understanding of concepts covered in MLTutor, performed better in terms of their test scores. The mean score of group 3, with the least computer experience and background knowledge, is lower than for the other groups. A summary of these results taken from D.2.2 is given in table 9.1 below.

	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
Mean	20.8	19.9	16.3	21.3	17.4	19.9
Stdev	4.4	4.6	4.3	2.6	2.3	5.8

Table 9.1: Summary of means and standard deviations of group scores from D.2.2

These results are as expected and support the decision to allocate a different version of MLTutor to each group member; thus minimising the likelihood of prior knowledge skewing the results in favour of any particular MLTutor version.

In the second phase of this analysis, the participant' scores were analysed based on usage of adaptive and non-adaptive versions of MLTutor. Means and standard deviations based on this evaluation criterion can be found in table D.3.2 and a summary of the results is presented in table 9.2 below.

	Non-Adaptive MLTutor	Adaptive MLTutor
Mean	17.9	19.6
Stdev	4.2	4.2

Table 9.2: Summary of means and standard deviations of scores from D.3.2.

These results show that scores using an adaptive version are slightly higher than for the non-adaptive version of MLTutor. The feasibility of performing further parametric tests on this data was discussed[1] and attempted. However, following subsequent discussion[2] it became clear that the sample sizes available were too small to conduct this type of analysis.

In the next phase of evaluation the participants' scores were analysed in terms of the MLTutor version used. This data can be found in table D.3.1 which is the data from table D.2.1 re-categorised based on MLTutor version. An analysis of this data showing means and standard

[1] With Dr. Kevin Boone, School of Computing Science, Middlesex University.

[2] With Dr. David Westley, School of Psychology, Middlesex University.

deviations, based on version, can be found in table D.3.2 and is summarised in table 9.3 below.

	Non-adaptive	Version 1	Version 2	Version 3	Version 4
Mean	**18**	**21.4**	**20.7**	**20.2**	**16**
Stdev	4.2	3.3	2.2	4.9	4.7

Table 9.3: Summary of means and standard deviations of scores from D.3.2.

The results indicate that the highest mean score was attained by users of MLTutor version 1 which was an adaptive version, and the lowest mean score was attained by users of MLTutor version 4 again an adaptive version. The second lowest mean score was attained by users of the non-adaptive MLTutor version.

Standard deviations for these results are quite high. It is believed that this is a consequence of each version being tested by participants with differing backgrounds which resulted in large variations in test scores for each version. The experiment was set-up in this way in order to eliminate any bias in the results due to prior background knowledge and computer experience.

In summary, these results are inconclusive and can only partly support the hypothesis that an adaptive version of MLTutor results in higher scores compared to the non-adaptive version. Although the results are suggestive, the number of participants who tested the variants of MLTutor is too small to apply any further parametric statistical analysis.

9.2.2 Analysis of feedback questionnaires

The participants in the evaluation were asked to complete a feedback questionnaire after using MLTutor to answer the set tasks. The feedback questionnaire consisted of six questions where the participants were asked to provide feedback on a scale, which was converted to a value between 0 and 9 to assist with the analysis. The participants' responses can be found in table D.7.1 of Appendix D. In addition to these six questions the participants were given the opportunity to provide free form textual feedback and this information is available in Appendix C.5.

9.2.2.1 Fixed format user feedback questions

Means and standard deviations based on the first part of the feedback questionnaire were calculated and the results can be found D.7.2. The findings are as follows:

For the question "How easy was the MLTutor system to use?" participants who used MLTutor version 1 gave the highest mean rating and achieved the highest mean test scores (see table 9.3) while participants who used MLTutor version 4 gave the lowest mean rating and achieved lowest mean test scores (see table 9.3). Users of the non-adaptive version gave the second highest mean rating to this question.

For the question "How frequently did you use the suggestion list?" participants who used MLTutor version 4 gave the highest mean rating, but gave the lowest mean rating in response to the question "Did you find the system suggestions relevant?". Log file analysis (see Appendix D.5.2) confirms users of this version to have used the suggestion list most frequently.

Participants who used MLTutor version 2 gave the highest mean rating for four of the six questions set in the feedback questionnaire. Furthermore, these participants achieved the 2nd highest average score overall (see table 9.3). However, their mean rating for the question "How would you rate the usefulness of the system suggestions?" was one of the lowest.

These results suggest that version 2 of MLTutor was overall the most highly regarded of the adaptive versions. Version 4 of MLTutor was least successful. Not only were the comments regarding the suggestions in version 4 adverse, the users of the version scored poorly.

Although the results for version 4 are particularly poor, it not clear whether this is due to the version itself or a consequence of the design of the experiment. Participants were allocated to a version on a first come first served basis and finding thirty volunteers to take part in the experiment was difficult. There is a suspicion regarding the commitment of some of the later volunteers who would have used version 4 due to the allocation scheme used.

9.2.2.2 Freeform user feedback questions

For each free form question on the feedback questionnaire, the comments of participants were examined with a view to identifying common themes and significant comments. The full text of participants' comments can be found in Appendix D. The common themes identified from comments made by users of the non-adaptive MLTutor were as follows:

- Unstructured information
- Not a tutoring system
- There is no inter-link between sites

The common themes identified from comments made by users of the adaptive variants of MLTutor were as follows:

- Easy to use
- Bookmarks are useful
- Some suggestions are good, but there should be more useful suggestions
- Suggests already visited pages
- Time delay before useful suggestions appear
- The entries in suggestion list were very difficult to understand
- A layman may need further background knowledge
- Provides a short-cut to relevant pages
- Made aware of links which were not apparent
- Offers different forms of navigational aid.

The comments made regarding the non-adaptive version of MLTutor, a plain HTML browsing system, tend to support the decision to develop the adaptive MLTutor.

Many user comments focused on the content of the sites within the MLTutor system. There were to the effect that the topics were well covered, up-to-date and the waiting time associated with the WWW was eliminated. Waiting time was reduced as, for the purpose of the experiment, the data was retrieved locally rather than over a network. As the research in this thesis is principally concerned with the adaptivity provided by MLTutor, and the Website content was prepared by others, comments regarding content were not considered significant.

It was suggested by a non-computer literate user that a layman may need further background information than that provided by this implementation of MLTutor. The MLTutor system in its current form is not a tutoring system but an adaptive information search tool that could be developed into, or used as a part of, a tutoring system.

In many cases, information on the WWW is unstructured and the adaptive features of MLTutor aim to overcome this by providing links between sites which are otherwise unconnected. Users liked the seamless integration of web-sites with a common interface and, as noted by participants, the adaptation provided by MLTutor provided them with additional choice while searching for information.

In addition, participants found the suggestion list entries a convenient shortcut to relevant pages that were otherwise not apparent or readily accessible. Although not heavily used the alternative form of navigation introduced by the suggestion list seemed to be appreciated.

A number of enhancements to MLTutor were suggested by participants as follows:

- Already visited pages can be re-suggested without indicating that they have already been visited. There is no history-based annotation within the entries of the suggestion list and adding this feature was suggested as an enhancement to the system. An option to hide already visited pages from the suggestion list might also be considered.

- For the research prototype implementation, the file names, as opposed to the page titles were used in the suggestion list. File names are not always as meaningful as the page titles, which should be used for any further development of the system.

- It was also suggested that the suggestion list should be keyword based allowing links to pages containing the keywords as opposed to the current implementation which lists pages and allows display of keywords covered on that page.

- The participants also suggested that the suggestion list should be permanently visible as opposed to being on a floating popup that can be obscured by the main browser window. They also commented that the content of the suggestion list should be more structured and an alternative representation, for example a graphical representation, would be useful.

- Users also commented on the time delay in the suggestion process which is due to the need to build up a list of pages to generate suggestions from and the periodic nature of the suggestion list refresh. Within MLTutor 10 pages have to be visited before suggestions can be made. This is due to the need to collect sufficient data for the machine learning algorithm to process. Once 10 pages have been visited, the suggestion list is refreshed periodically. The time between refreshes is set within the application and can easily be adjusted.

These comments from users of the system are instructive and suggest areas where effort should be directed for any future development work with the system.

9.2.3 Analysis of log files

Aspects of the participants' browsing activity were recorded in a log file in the background

during the experiment. The usage of three types of links and the task completion time were assessed from these log files.

The types of links examined are built-in links, suggestion list links and Website bookmark links. Link usage and total task completion time from each log file can be found in table D.4.1 in appendix D.

A quantitative data analysis was applied to this data. The primary objective of this particular analysis is to investigate the benefit of using adaptive MLTutor over non-adaptive MLTutor in terms of the criteria given below and to see if any of the adaptive versions consistently produced better results. The first two criteria below have been used by a number of other researchers (see chapter 8) and a secondary objective of this analysis is to create common grounds to allow a comparative study with the relevant research (see §8). For this analysis the evaluation criteria are:

Total task completion time: Total task completion time was analysed in order to investigate if any version, either non-adaptive or one of the adaptive variants, of MLTutor resulted in a shorter total task completion time.

Total usage of links during task completion: Total link usage was analysed in order to investigate if any version, either non-adaptive or one of the adaptive variants, of MLTutor reduced the total number of links used during task completion.

Total usage of suggestion list links during task completion: Total usage of suggestion list links used during task completion was analysed in order to see if suggestion list usage was higher in a particular variant of MLTutor.

Total usage of bookmarks during task completion: Total usage of bookmarks during task completion was analysed in order to see if bookmark usage was higher in a particular variant of MLTutor.

The findings of the analysis can be found in table 9.4.

Version	Mean	Time spent	Links visited	Bookmarks visited	Suggestions visited
0	Mean	73.2	111.2	8.3	Facility not available
	Stdev	24.6	49.2	8.1	
1	Mean	82.2	85.7	7.7	4.3
	Stdev	17.8	28.8	4.7	4.6
2	Mean	53.7	82.8	8.2	4
	Stdev	18.4	55.1	5.4	4.7
3	Mean	53.2	85	5.2	2.8
	Stdev	23.1	34.1	1.0	2.6
4	Mean	56.3	71.3	7.3	8.5
	Stdev	14.6	16.8	7.6	5.4

Table 9.4: Mean and standard deviation analysis of evaluation criteria.

The results suggest that the mean total task completion time was reduced for users of the adaptive versions, as was the total number of links visited. Within the adaptive versions the lowest mean time spent was achieved by users of version 3 and the lowest total link usage with version 4. Lower bookmark usage was also achieved with the adaptive versions, however only slightly, with the exception of version 3 users who had on average the lowest usage. Suggestion list link usage was also lowest for version 3 of the system and highest for version 4. The results for version 4 bear out the user feedback comments of §9.2.2.1.

Since the results, on the basis of the criteria above are promising, further parametric tests were conducted but did not produce statistically significant results, however, as previously stated the small sample sizes make these tests invalid and for this reason the results are not included.

9.2.4 Log file and participant score cross analysis

In addition to the analysis described in the previous sections, further in-depth analysis was conducted in order to investigate the usage of built-in and suggestion-list links. The objective of this analysis was to determine how frequently built-in links and MLTutor system suggestions were followed by the participants and utilised to complete the given tasks.

For this analysis, four categories of link usage were established as follows:

X1: If a user accessed the expert recommended page by a built-in link and answered the question correctly.

X2: If a user accessed the expert recommended page either by a built-in link or by a suggestion list link and answered the question correctly.

O1: If a user accessed the expert recommended page by a built-in link and failed to answer the question correctly.

O2: If a user accessed the expert recommended page either by a built-in link or by a suggestion list link and failed to answer the question correctly.

Ideally, six categories would have been preferred allowing built-in and suggestion list links to be analysed individually, however, the data available from log files was not sufficient to allow this and so usage of these link types were merged to form categories X2 and O2.

In order to assist with this analysis, a *link usage analysis form* was created (see table 9.5). Part 1 of the form contains details of the participant to whom the form refers. Part 2 of the form gives a count of the number of times a suggestion list link was used to access the pages listed in part 4. Part 3 of the form contains counts of the number of times a built-in link was used to access the page numbers in part 4 of the form. Part 4 of the form lists the pages which contain the expert recommended model answers. In part 5 of the form there is a row for each of the set tasks. Within each of these rows, the pages containing the model answers to the question the row corresponds to, are highlighted in dark grey. For example pages 63 and 127 are the recommended pages to answer question 4.

This form is populated with data from various sources. The log file of each participant was analysed in conjunction with the answer sheet completed by the participant. Each answer was rated in terms of the four categories defined above and transferred to part 5 of the form. Data to complete part 2 and part 3 of the form was also extracted from log files and for convenience is tabulated separately in tables D.5.2 and D.5.1 respectively.

A complete *link usage analysis form* was created for each participant and these can be found in Appendix D. For convenience, the columns containing suggestion link usage are highlighted in light grey in these forms.

1	Participant 20Group: 2Version:																														
2	Suggestion-list Links																														
3	Built-in Links																														
4	Expert rec. Links	2	5	12	61	62	63	74	77	87	88	94	95	101	104	109	110	112	114	115	116	119	121	122	123	126	127	128	129	130	132	133
5	Question 1																														
	Question 2																														
	Question 3																														
	Question 4																														
	Question 5																														
	Question 6																														
	Question 7																														
	Question 8																														
	Question 9																														
	Question 10																														
	Question 11																														
	Question 12																														
	Question 13																														

Table 9.5: A *link usage analysis form*.

From an examination of the data gathered during the experiment, it became apparent that the pages containing answers to the questions had been accessed via various types of links i.e. suggestion list and built-in. Further, the pages containing the answers had been repeatedly accessed which had not been anticipated prior to the experiment. It was thus not clear which type of link had been used to answer the set tasks. In many respects this highlights a flaw in the design of the experiment, as the time the tasks were answered was not time stamped or audited. This would not be feasible with the paper-based format of the task completion which was used in this experiment. An on-line exercise, as suggested by one of the participants on the feedback questionnaire would be a solution to this problem.

The total occurrences for the categories X1 and X2 were summed from the *link usage analysis forms* and transferred to table 9.6.

	1st Participant	2nd Participant	3rd Participant	4th Participant	5th Participant	6th Participant
Version 1	X1= 13 X2= 0	X1= 7 X2= 3	X1= 6 X2= 3	X1= 10 X2= 1	X1= 15 X2= 0	X1= 18 X2= 0
Version 2	X1= 12 X2= 3	X1= 6 X2= 0	X1= 7 X2= 2	X1= 9 X2= 0	X1= 7 X2= 0	X1= 10 X2= 0
Version 3	X1= 8 X2= 0	X1= 10 X2= 3	X1= 12 X2= 1	X1= 11 X2= 3	X1= 10 X2= 0	X1= 11 X2= 0
Version 4	X1= 4 X2= 0	X1= 7 X2= 3	X1= 2 X2= 7	X1= 15 X2= 0	X1= 3 X2= 8	X1= 6 X2= 2

Table 9.6: Summary of the X1 and X2 categories in the *link usage analysis form*.

Within the population of participants, across the adaptive variants of MLTutor, the total occurrences of categories X1 and X2 are equal to 219 and 39 respectively.

The percentage of correct answers where the answer page has been visited by built-in links only is $\frac{X1}{X1+X2} = \frac{219}{258} = 85$ and the percentage of correct answers where the answer page has been visited using only suggestion list links is less than $\frac{X2}{X1+X2} = \frac{39}{258} = 15$. These results indicate that the preferred navigational method for completing tasks was to follow available built-in links. Although the results of the statistical analysis are indicative of the benefits of adaptivity (see §9.2.3), the log file and participant score cross analysis contradicts with this evidence. The quantitative evaluation criteria used in the analysis, also commonly used to measure the effectiveness and efficiency of adaptation in adaptive hypermedia research, on their own, fail to reflect the actual usefulness of the adaptation. The results indicate that the use of an adaptive feature does not prove anything unless related to improved performance in some way. The enhanced results alone may simply be the consequence of an alternative interface and not directly a consequence of any adaptation.

The results of the empirical study are revisited in Chapter 10, the next section describes the technical evaluation applied to MLTutor.

9.3 Technical evaluation of adaptive MLTutor

In this technical evaluation of MLTutor, the performance of clustering within MLTutor was investigated. The clustering phase of the machine learning component is largely responsible for

the quality of the suggestions made and adverse user feedback was focussed on this issue. Two aspects of the clustering have been investigated and are discussed below.

9.3.1 Sort step sensitivity

The conceptual clustering algorithm (Hutchinson 1994) employed by MLTutor contains a sort step which orders pairs of pages based on the distance between them. If two pairs of pages are equally far apart they can legitimately appear in any order following the sort. As commented by Hutchinson (1994) and demonstrated in Appendix E.1 this can have a significant impact on the clustering.

The decision to use integer values to encode the page attribute descriptions and the metric used to measure the distance between pages leads to the likelihood of there being a high incidence of equally distant pages.

The impact of this in terms of the resultant clustering in MLTutor is hard to determine from the available information but could have had an impact on the clustering and ultimately the suggestions made by MLTutor.

9.3.2 The 'bin' cluster

For the implementation of pre-clustering in versions 3 and 4 of MLTutor (see chapter 7), all 133 pages available within the system were clustered applying the same algorithm used in the dynamic variants of MLTutor. This resulted in 16 clusters being formed. Running the clustering algorithm on 133 pages initially generated 10 clusters, however the 10^{th} cluster contained 103 pages and the clustering algorithm was re-applied to this data. The second phase of clustering produced three clusters and yet again another quite large (final) cluster. The re-clustering process was repeated until no further subdivisions could be achieved. Having repeated the clustering process six times 16 clusters were created (see appendix E.8). The 16^{th} cluster was still very large, containing 65 pages.

During the pre-clustering process it was observed that each re-clustering attempt produced one heterogeneous cluster – which in each case appeared to be the largest cluster – that contained a combination of items which seemed to be semantically unrelated. This thesis refers to this last and largest cluster as the 'bin' cluster. The 'bin' cluster has implications for both dynamic clustering and pre-clustered versions of MLTutor and the effect can be observed in the users' log files.

In the case of dynamic clustering, if the cluster selected for inducing rules is the 'bin' cluster then it is very likely that any suggestions produced for this cluster would be unfocused and cover most of the pages available within the system. This problem was not appreciated when the MLTutor system was tested on 32 pages (see chapter 7). Although the initial evaluation highlighted a number of issues these were not attributed to the clustering algorithm and tests with the 32-page document and alternative cluster selection strategies produced promising results.

In the pre-clustering case the 'bin' cluster 16 contains 65 pages out of 133 and there is a greater likelihood of the last or the most currently visited pages being in this cluster. The consequence of this is that the user is presented with many unfocused suggestions.

Although it is possible to devise a solution to overcome the problem with dynamic clustering, for example by not suggesting anything if the most recent page is in the 'bin' cluster, this would not be feasible in the MLTutor versions which employ pre-clustering. Discarding the 'bin' cluster from the adaptation process means nearly half of the hypertext pages in MLTutor would not be capable of triggering suggestions. An alternative solution would be to manually pre-cluster the data. This would be time consuming and may require expert guidance.

In summary, any adaptation provided on the basis of the 'bin' cluster is low quality and unfocused. Given these issues, an appropriate solution may be to replace the clustering algorithm with one more sensitive to the needs of MLTutor.

9.4 Conclusion

The aim of the research presented in this thesis has been to determine the viability of utilising machine learning techniques to provide adaptation in a hypertext environment.

The framework proposed in Chapter 8 was developed in order to evaluate the effectiveness of the application of machine learning in MLTutor. The results of the evaluation based on this framework were presented in this chapter.

The results of the analysis, in many respects, are inconclusive. Many of the findings (see §9.2.3) based on the quantitative measures are in favour of an adaptive version of MLTutor. However, the *log file and participant score cross analysis* has shown that quantitative measures fail to provide conclusive evidence that the use of the adaptation results in beneficial improvement.

Similarly, feedback and opinions from participants may be misleading when not used in context or related to their actual interactions.

In the next, concluding chapter, these results are revisited and recommendations for further work made.

Chapter 10

Discussion and Conclusions

10.1 Introduction

The research presented in this thesis is reviewed in this chapter. Following the review, the key issues raised by the research are discussed and conclusions drawn. The relevance of the research to the contributing domains is stated and this is followed by recommendations for further work and concluding remarks.

10.2 Summary of the thesis

A problem, which commonly faces hypertext users, is the difficulty of identifying pages of information most relevant to their current goals or interests. Consequently, they are forced to wade through irrelevant pages even though they know precisely what they are looking for.

In order to address this issue, this research has investigated the technical feasibility and also the utility of applying machine learning algorithms to generate personalised adaptation on the basis of browsing history in hypertext. To demonstrate the viability of this approach, a Web-based information system called MLTutor was designed, implemented, tested, and evaluated.

The MLTutor design aimed to remove the need for pre-defined user profiles and replace them with a dynamic user profile-building scheme in order to provide individual adaptation. This is achieved in MLTutor by employing a combination of clustering and rule induction algorithms.

The clustering algorithm within MLTutor is used to classify Web pages accessed by a user with a view to identifying commonality in the browsed pages. The rule induction algorithm is used to generate classification rules, which are then used to provide adaptive navigational support.

The adaptive navigational support provided by MLTutor is in the form of a list of system suggested Web pages. The decision to suggest a page is based on the rules generated by the rule induction algorithm. The entries in the suggestion list are annotated; selecting an entry from the list displays keywords associated with the page, and the page itself can be selected.

The MLTutor has been implemented as a Web-based client server system. The client component allows Web pages within the system to be viewed with a standard Web browser. The client records the browsing history and controls display of the system suggestions. The suggestions themselves are generated by the server-based machine learning component of the system when requested by client.

Within the machine learning component attribute based machine learning algorithms have been implemented. Attribute based algorithms process objects, in this case pages, described in terms of a number of specified attributes. Following several experiments, a scheme for selecting attributes for use in MLTutor was devised, based on a classification of hyperlink anchors within the Web pages of the system.

A simple conceptual clustering algorithm (Hutchinson 1994) was used for the MLTutor clustering process. For the rule induction phase the ID3 algorithm (Quinlan 1986) was initially used. However, it was found not to be ideally suited to this application. Consequently, the SG-1 algorithm, an enhancement of ID3, was developed.

In order to evaluate the benefits of the machine learning approach, a comparative study was conducted using five versions of MLTutor including one with the adaptivity disabled. Two of the adaptive variants applied the clustering algorithm dynamically and used two different cluster selection strategies. These strategies were based on last page visited and a weighting of recently visited pages. The other adaptive variants used pre-clustered data with the same cluster selection strategies.

The aim of the comparative study conducted on MLTutor was to determine if the adaptive versions of the system allowed system users to perform tasks more successfully than users of the non-adaptive version and, if so, was any adaptive version more successful than others. In order to perform the comparison an evaluation framework was established using several measures.

The principal measure used in the evaluation was the answers to questions completed by participants while using a version of MLTutor. Additionally, link usage and time taken to complete the exercises were used. An original analysis, called the *log file and participant score cross analysis*, to determine how participants reached the pages containing answers to questions was also undertaken.

In terms of test scores achieved by the participants, the users of the adaptive versions scored on average higher than the users of the non-adaptive version. Within the versions themselves there was no clear favourite: users of three of the adaptive versions scored on average better than users of the non-adaptive version, while users of one of the adaptive versions scored particularly poorly.

This analysis highlighted a number of issues with regard to the design of the experiment. There is some suspicion that the motivation and commitment of the participants may have played a part in the poor results achieved for this version. While all the participants in the evaluation were volunteers, a number of the later volunteers may have felt obliged to take part and consequently have been less motivated than the earlier volunteers. Unfortunately, due to the method of allocating subjects to versions the later volunteers would have all used the same version of the system: the adaptive version with the poorest test scores.

Although an in-depth parametric statistical analysis could not be conducted on the quantitative data gathered during the experiment, the participants' test scores and other measures e.g. link usage and the total time to complete the exercises, used in the evaluation of MLTutor tended to support the adaptive versions. However, the findings based on the *log file and participant score cross analysis* contradict this claim. The quantitative measures, on their own, fail to reflect the actual usefulness of the adaptation. The results of the latter analysis indicate that the use of an adaptive feature does not prove anything unless related to a user's improved performance in some way.

Beyond the quantitative data gathered during the experiment, qualitative data was also collected in the form of feedback comments from the participants. Not all feedback from the participants who took part in the evaluation was positive.

While participants seemed to like the idea of the seamless integration of Web-sites with a common interface, and appreciated the alternative form of navigation introduced by the suggestion list, which they saw as a convenient shortcut to relevant pages that were otherwise not apparent or readily accessible, they also raised a number of issues with the design, some of which are more fundamental than others.

The issues raised largely focussed on the format and structure of the suggestion list. Within the current implementation of MLTutor, file names, as opposed to the page titles, were used in the suggestion list. File names are not always as meaningful as the page titles, which were found to be confusing at times.

Further, there is no history-based annotation within the entries of the suggestion list and already visited pages can be re-suggested without indicating that they have already been visited. This problem was probably exacerbated by the use of unfamiliar file names in the suggestion list.

The participants also commented on the time delay before suggestions were made. This is initially due to the need to build up a list of pages to generate suggestions from and not something easily remedied with the current design. The only input available to the machine learning component of the MLTutor are browsed pages. Given that the MLTutor explicitly aimed to remove the need for a registration process requiring users to state their objectives, making suggestions before sufficient data has been collected would require assumptions to be made about the motivations of the user. Following creation of the first suggestion list, after ten pages have been visited, the suggestion list is periodically re-built in the background. For the purposes of the evaluation this was approximately every four minutes but can be adjusted to an alternative period if required.

10.3 Contributions

This research is multidisciplinary and is an attempt to bridge the gap between theory and practice in the domains of machine learning and WWW-based adaptive hypertext. This research offers contributions in two principal areas, machine learning and adaptive hypertext.

10.3.1 Contributions to machine learning research

A novel approach in machine learning

The MLTutor prototype uses a combination of clustering and inductive machine learning algorithms. This machine learning approach facilitates a flexible, individualised approach to adaptation within MLTutor without the need for additional input beyond browsing patterns. This integration of two machine learning algorithms is a novel approach in the field of machine learning.

The SG-1 multiple decision tree building machine learning algorithm

The ID3 algorithm was initially used for the rule induction process in MLTutor. Although this algorithm has many practical applications, and has been used in a number of systems, it was found not to be ideally suited within MLTutor.

ID3 is effective in generating efficient decision procedures and is suited to situations where a single best procedure is required. However, ID3 does not take into account alternative hypotheses that are consistent with the available data when constructing decision procedures.

In the context of MLTutor this was found to be a weakness. Rule induction within MLTutor is

used to produce suggestion rules describing clusters generated from browsed input, and all hypotheses consistent with the clustered data are of interest when suggesting related pages in MLTutor.

In order to address this weakness the SG-1 algorithm was developed as an enhancement of the ID3 procedure. SG-1 maintains all hypotheses consistent with the training data as with the focusing algorithm discussed in §5.2.2. However, in contrast to the focusing algorithm, SG-1 does not have any problem representing disjunctive concepts. Rules produced by SG-1 are based on all hypotheses and so suggestions based on these rules better reflect the area of interest contained within browsing patterns.

While the SG-1 algorithm has been developed specifically to address the needs of MLTutor, its application is not restricted to use in MLTutor. As an addition to the family of supervised decision tree building algorithms, there is scope for application in other areas where multiple concept descriptions need to be maintained.

The 'bin' cluster

As discussed in §9.3.2, in the pre-clustering process, a series of steps were applied to classify the document within MLTutor (see Appendix E). Each pre-clustering step produced a large heterogeneous cluster containing semantically unrelated hypertext pages referred to as the 'bin' cluster.

The effects of using the 'bin' cluster were observed during evaluation. In the case of dynamic clustering, if the cluster selected for inducing rules was the 'bin' cluster then any suggestions produced for this cluster were unfocused. In the pre-clustering case the 'bin' cluster contained almost half of the pages within the system and there is strong chance that the last, or the most recently visited pages will be in this cluster. The consequence of this is that the user is presented with almost half of the pages within the system as suggestions.
This problem was not appreciated when the MLTutor system was tested on a small document. Although the initial evaluation highlighted a number of issues, these were not attributed to the clustering algorithm, and tests with alternative cluster selection strategies produced promising results.

Although it is possible to devise a solution to overcome the problem when using dynamic clustering, for example by not suggesting anything if the most recent page is in the 'bin' cluster, this would not be feasible with pre-clustering. In this case discarding the 'bin' cluster from the

adaptation process means nearly half of the hypertext pages in MLTutor would not be capable of triggering suggestions. An alternative solution would be to manually pre-cluster the data. However, this would be time consuming and would require expert guidance.

10.3.2 Contributions to adaptive hypertext research

To determine how the adaptive component of MLTutor was utilised to complete a set of tasks, an original *log file and participant score cross analysis* was developed. During this cross analysis a number of weaknesses related to the evaluation methods used in the field adaptive hypermedia were determined. These are:

- The use of quantitative measures, on their own, may fail to reflect the actual benefits of adaptation.

- The use of an adaptive feature may not prove anything unless related to the improved performance in some way.

In terms of MLTutor, although the results of the statistical analysis are indicative of the adaptivity, the *log file and participant score cross analysis* indicates that the adaptivity available within the system has hardly been used. By simply measuring differences in performance between adaptive and non-adaptive versions of a system, a fundamental assumption that the difference is due to the adaptivity is made. However, the cross analysis conducted as part of this research suggests that this assumption may not be valid.

10.4 Key features

Dynamic user modelling

The ideal adaptive system is described by Brusilovsky (1996) as follows: " ... while the user is simply working in an application system, the adaptation component watches what the user is doing, collects the data describing user's activity, processes these data to build the user model, then provides an adaptation. Unfortunately, such an ideal situation is very rarely met in adaptive hypermedia systems...".

Adaptive systems often fail to meet this ideal due to the need to interrogate users about their requirements or interests. This can be in the form of an initial registration process where user needs are assessed, or during interaction with the system in the form of questions or the

provision of relevance feedback. The information gathered in this process is typically used to allocate a pre-defined stereotypical profile to a user, which is used to control adaptation.

There is an overhead in generating such stereotypical profiles and a danger that an inappropriate one may be selected for a user. By basing adaptation in MLTutor on browsing history, no additional feedback is required from a user and by using machine learning techniques to dynamically analyse the browsing history there are no requirements for pre-defined profiles to be constructed. In this respect the MLTutor comes close to Brusilovsky's ideal system.

Browsing path analysis

Balabanovic (1997) points out that a machine learning approach to support browsing without any particular goal in mind is useless. MLTutor has been designed for use in an educational context and specifically supports task-oriented browsing. If the browsing activity of a user is aimless, it is difficult to determine any regularity or any meaningful pattern in the browsing behaviour. However, if the intention of a user's navigation is to complete a number of specific tasks, or to answer questions during browsing, then monitoring the user interaction in order to provide guidance becomes more feasible (Taylor and Self 1990).

MLTutor uses machine learning techniques to search for patterns within the content of material accessed during a user's information seeking activity. Beaumont (1994) argues that the bandwidth of the information contained in a user's browsing pattern might be too narrow to elicit information about the user' interest; however, as stated by other researchers (Hirashima 1998; McEneaney 1999), browsing patterns are a fundamental source of information representing the user's interaction with the system. While such patterns have been investigated by several researchers (Sun *et al* (1995); Lieberman (1995); McEneaney (1999)) who have applied various AI techniques e.g. heuristic search, dynamic programming, neural networks, little work has been done on the application of machine learning techniques to dynamically build user profiles in the field of adaptive hypermedia.

Within MLTutor, any patterns identified in browsing history by the machine learning component of the system are used to dynamically generate suggestion rules which are used to recommend pages related to the browsing. By this mechanism the MLTutor system aims to help users to locate information relevant to their current interest.

The suggestion rules within MLTutor are generated by use of a novel combination of clustering and inductive machine learning algorithms. The dynamically generated rules from a profile hold

a generalisation of the user's current area of interest. This profile is updated as further pages are visited and new rules created.

Attribute based systems

Information retrieval systems rely on document categorisation strategies and these strategies are typically based on keyword descriptions of information. These keyword descriptions are used by information retrieval systems to relate associated documents together. Similarly, attribute based machine learning algorithms process attribute descriptions of objects. In both cases the selection of features to describe objects is fundamental to performance.

The MLTutor has based attribute encoding on a classification of hyperlink anchors within the hypertext. Hypertext links facilitate navigation and as such typically relate pages at a conceptual level. A distinct advantage of using hypertext anchors to describe documents is the ease with which they are identified within the hypertext.

Although this strategy relies on hypertext documents being constructed in a sensible manner the strategy has been shown as a viable option. However, the MLTutor design is capable of incorporating any attribute based scheme for encoding documents due to the nature of machine learning algorithms employed.

Adaptive navigational support

Adaptive hypertext techniques fall into two categories, namely presentational and navigational. By far the most researched of these are navigational support techniques. The annotated suggestion list within MLTutor falls into this category.

As a result of the machine learning process within MLTutor the suggestion list is adaptive to the ongoing and changing requirements of the user. This is achieved by use of a sliding window technique that takes into account recent browsing. The use of browsing paths alleviates the need for any additional feedback from a system user at all.

Furthermore, the adaptivity provided by MLTutor can be disregarded without penalty or overhead to the user. By ignoring system suggestions, or turning the facility off, the MLTutor becomes a plain hypertext browsing system familiar to all users of the WWW. This solution to providing adaptivity is intended to prevent alienation of potential users; some users may not need to see suggestions and others may not be comfortable with unfamiliar adaptive features within the interface.

WWW based systems

While the WWW offers huge potential for distance learning, the mechanism of the Web can be employed to deliver information within an individual organisation or classroom.

The WWW can be considered a vast network of interlinked documents, effectively a huge hypertext system. Within this network specific sites cater for specific needs often with links to other related sites. However, the Web of documents is growing in an unregulated and unstructured manner with important connections between highly related documents missing. Consequently finding specific or relevant information on the Web can be difficult.

The MLTutor presents a solution to this issue by introducing an adaptive facility, which aims to assist in the search for information. The MLTutor dynamically generates an individual profile in the form of suggestion rules based on a user's history of browsing activity. These rules are used to suggest additional pages the user may be interested in. A significant benefit of the MLTutor is that suggestions may relate to documents the user has not yet seen, or may not be aware of, as they are not directly connected to the document the user has accessed. The suggestion list mechanisms in MLTutor allows direct access to these additional pages even though no link physically exists between them.

The machine learning approach employed to build the list of suggestions ensures that the suggestions made are relevant to the user's current area of interest. Effectively, the MLTutor provides a mechanism for re-structuring a hypertext document to cater for individual preferences without restricting access in any way.

This approach has synergies with the approach suggested by Stotts and Furuta (1991) who proposed a flexible structure, to overlay a fixed structure, as a solution to personalising a hypertext system.

Information overload and lost in hyperspace

There are many advantages associated with hypertext systems. However, as described in Chapter 3 there are associated disadvantages. There problems fall into two broad categories, information overload and user disorientation. Disorientation results when users find themselves lost in the hyperspace of information and overload occurs when excessive information, not of direct relevance, is presented.

The MLTutor has been designed to provide information related to a user's area of interest by

analysing browsing patterns and suggesting pages of information related to this browsing. Consequently the MLTutor attempts to reduce the possibility of information overload by filtering out irrelevant pages.

However, MLTutor only partially addresses the problem of user disorientation. Elm and Woods (1985) state three common causes of user disorientation (§ 3.6): users not knowing where to go next, knowing where they want to go but not knowing how to get there and users not knowing where they are within the overall structure. By presenting a list of related topics, the MLTutor suggestion list has the potential to assist users who know where they want to go but not how to get there and may assist users who do not know where to go next.

It is recognised that MLTutor introduces additional options for next page selection and the current implementation does not incorporate any features to assist users who do not know where they are within the overall structure; however the bookmark feature at least allows users to relocate themselves to known points.

The ability of MLTutor to assist with user disorientation and information overload is critically dependent on the quality of the suggestions made and the confidence users have in the suggestions. Without confidence additional facilities provided by MLTutor could be seen as distracting. However, the system contains the facility to turn off the suggestion list if this is the case.

10.5 Future work and enhancements

A number of areas where further work will be required to develop MLTutor from a prototype to a fully functional WWW-based system have been identified. These are discussed below.

Automating the addition of documents to MLTutor

The attribute database used in the prototype versions of MLTutor has been hand built. However, the principles used to construct the database, based on the use of hyperlink anchors (see §7.3.1), make automation of this process feasible.

An automated process would be beneficial, as attribute database construction time would be greatly reduced allowing additional documents to be integrated into the system more rapidly.

Options for automating the attribute generation process were briefly discussed in §6.2.4. These

approaches focussed on the use of *meta* data embedded within the content of the system using standard features of the HTML language used for Web page construction. However, any scheme for extracting attributes from a document can be used.

Once such a scheme has been established, additional documents can be added to MLTutor with relative ease.

- The keyword catalogue will need to be reviewed to take account of any new keywords added due to the new pages.

- The attribute database will need to be reconstructed. One record for each available page will be created indicating presence or not of keywords on the page.

- A page applet will need to be added to the new pages (see §6.3) associating the page to the attribute database entry.

A program to achieve the above should be relatively straightforward to develop.

Although use of an automated process looks feasible, the assistance of an expert in the domain of the content would still be advisable to review the results. However, this would still reduce the effort required to publish documents within MLTutor and all technical details could be hidden behind a user-friendly interface removing the need for technical knowledge of how the system works.

Alternative cluster selection strategy

In the current implementation of the MLTutor prototype, two different cluster selection strategies are used. These are based on the last page visited and a weighting of recently visited pages. The impact of alternative cluster selection strategies should be investigated. One alternative strategy would be to base cluster selection on the determinant ratio for the cluster. The determinant ratio is a measure of the cluster quality, however, this does not take into account the size of the cluster. In table 7.2 clusters two and three have the highest determinant ratios but only contain a small number of pages. Cluster one has the second highest determinant ratios but contains more pages. The investigation into using determinant ratio as the basis for cluster selection will need to consider cluster size along with the determinant ratio.

Alternative clustering algorithms

The technical issues with the clustering algorithm will need to be addressed before further development of MLTutor takes place. The most appropriate solution may be to replace the simple conceptual clustering algorithm with an alternative clustering strategy. One alternative worthy of consideration is COBWEB (Fisher 1987). COBWEB builds groupings of attribute based objects in a bottom-up manner, as does the simple conceptual clustering algorithm used in MLTutor. A second option worthy of consideration is AutoClass (Cheesemann *et al* 1988). AutoClass is an unsupervised Bayesian classification system, which produces a set of class descriptions from a set of attribute vectors.

Although these machine learning programs are likely candidates for replacing the simple conceptual clustering algorithm in MLTutor, no work other than a brief survey of available options has been undertaken to date.

Implementation on the WWW

Although developed as a WWW-based system, for the purpose of investigation and evaluation the MLTutor prototype has not been tested in a networked environment. Rather it has been tested on a workstation with a (personal) Web server installed locally. This arrangement mirrors the networked environment closely and consequently porting MLTutor should be simple. However, the nature of the testing environment means that performance in a Web-based environment with concurrent user access has not been established. The design of MLTutor makes use of Internet technology though, and it is strongly believed that no significant problems will be encountered.

Format of the suggestion list

Feedback from the MLTutor prototype evaluation indicated several possible enhancements to the format of the suggestion list.

With hindsight the decision to use HTML file names in the suggestion list, as opposed to page titles, was flawed. Page titles are much more meaningful than filenames and making this change would greatly enhance the usefulness of the suggestion list with minimal effort.

Although more complex to implement, it was also suggested that a graphical representation, rather than a list, would enhance usability. Implementing the system suggestions as a map would allow users to more easily locate themselves within the hypertext. Fisheye views and mapping techniques in hypertext were discussed in §3.2 as a solution to the lost in hyperspace

problem and consequently a graphical representation should be considered for MLTutor. However, further investigation would be required to assess the benefit of this approach.

A more significant change to the format of the suggestion list also merits further investigation. In its current form the suggestion list displays pages suggested by the system and the suggestion list annotation feature allows keywords associated with a suggestion page to be displayed. As an alternative, the utility of displaying a list of keywords related to the recent browsing should be considered, with selection of a keyword from the list resulting in pages associated with the keyword being made available for selection. The participants of the MLTutor evaluation who suggested this approach believed that this would have helped them to find information more easily than with the page based format of the suggestion list. Again further investigation to assess the benefits of this approach, which could be implemented as all the required information is available, would be required.

10.5 Synergy with recent research

Since work on MLTutor commenced, the idea of analysing information requests as a primary source of data has become more widespread and the work of several researchers (Crabtree and Soltysiak (1998); Pazzani and Billsus (1997)) has a strong synergy with the research presented in this thesis.

The use of clustering techniques to learn a user profile from a collection of documents has been investigated by Crabtree and Soltysiak (1998). In their system a user's word processing documents, emails and Web browsing activities are monitored to build an interest profile. Highest information bearing words are extracted from these documents to create vectors which are then clustered. These clusters are presented to users for feedback regarding relevance to their interests.

The clustering technique used in this system is a top-down statistical approach. Crabtree and Soltysiak (1998) anticipate that the clusters produced by the system could be used for personalised information retrieval and filtering tasks. An overriding aim of this work was to reduce, as much as possible, the need for a user to provide input to the system.

As with MLTutor the monitoring conducted by this system is unobtrusively performed and keyword based page descriptions are used to cluster the accessed documents to learn user interest. The system aims to reduce the need for users to provide feedback but does not

completely eliminate this.

The Syskill and Webert (Pazzani *et al* 1997) system has similar objectives. The Syskill and Webert browsing assistant asks users to rate Web pages as interesting or not and learns a user profile which is used to suggest other pages of possible interest. One profile per topic is learned which is much more specific than a generic user profile could be. Several machine learning algorithms were used for profile learning including ID3 and C4.5, however no algorithms produced better results than a Bayesian classifier (Pazzani *et al* 1997).

As with MLTutor this system used machine learning techniques to build user profiles but relied on users rating documents, which are used as training data for the algorithms.

10.6 Conclusion

This chapter has presented a high level review of the work undertaken on the MLTutor development to date. The research undertaken has covered a number of disciplines and the contributions to these areas were stated. This chapter concluded by indicating areas where further development is required and stated synergies with other recent research.

The objective of this research was to design, implement, test and evaluate a prototype system capable of demonstrating the technical feasibility of using machine learning techniques to analyse browsing patterns within hypertext, and to use this analysis to provide adaptive navigational support without the need for pre-defined stereotypical profiles.

The MLTutor system was developed to meet these objectives. MLTutor uses a combination of machine learning algorithms and does not require any pre-defined profiles or additional input beyond browsing patterns to build a personalised list of pages related to the browsed input.

In order to evaluate the system a comparative empirical study was conducted. The evaluation of adaptive systems is particularly complex as the results of the adaptation are personal to a specific user's set of circumstances and, as such, an empirical study is the most appropriate strategy for evaluation.

The importance of the design of an empirical study cannot be understated, and a number of issues with the design of the MLTutor experiment became apparent during analysis of results. Although faults in the experimental design limit the conclusions that can be drawn about

MLTutor, the results of the evaluation do show that MLTutor is a robust and functional system and suggest the potential benefits of using a machine learning approach to provide adaptivity based on an analysis of browsing patterns.

References

References

Akscyn R., McCracken D. and Ynder E. (1988) 'KMS, a Distributed Hypermedia System for Managing Knowledge in Organisations' *Communications of ACM*, Vol. 31 No. 7 pp 820-835, July 1988.

Anderson J.R. (1983) 'The Architecture of Cognition' *Harvard University Press*, Cambridge, Massachusetts.

Anderson J. R. and Reiser B. J (1985) 'The Lisp Tutor' *Byte*, Vol. 10, pp 159-75.

Anderson J.R. (1988) 'The Expert Module' Polson M.C. and Richardson J.J. (eds) *Foundations of Intelligent Tutoring Systems*, pp 21-53.

Armstrnng R., Freitag D., Jnachims T. and Mitchell T. (1995) 'WebWatcher: A Learning Apprentice for the World Wide Web' *AAAI Spring Symposium on Information Gathering from Distributed, Heterogeneous Environments*, Stanford, CA.

Baecker R.M., Grudin J., Buxton W.A.S., Greenberg S. (1995a) 'Hypertext and Multimedia' *Human-Computer Interaction: Toward the Year 2000*, Chapter 13, pp 833-842.

Baecker R.M., Grudin J., Buxton W.A.S., Greenberg S. (1995b) 'From Customizable Systems to Intelligent Agents' *Human-Computer Interaction: Toward the Year 2000*, Chapter 12, pp 783-791.

Balabanovic M. (1997) 'Exploring versus Exploiting when Learning User Models for Text Recommendation' *User Modeling and User-Adapted Interaction*, pp 71-101, Kluwer Academic Publishers.

Beaumont I. (1994) 'User Modelling and Hypertext Adaptation in the Tutoring System ANATOM-Tutor' *UM'94 Fourth International Conference on User Modelling.*

Benyon D., Stone D. and Woodroffe M. (1997) 'Experience with developing multimedia courseware for the World Wide Web: the need for better tools and clear pedagogy' *Int. J. Human-Computer Studies*, 47, pp 197-218.

Berners-Lee T. (1992) 'An Architecture for Wide Area Hypertext' *Hypertext'91 Poster Abstract, SIGLINK Newsletter*, December 1992.

Boyle C. and Encarnacion (1994) 'Metadoc: An Adaptive Hypertext Reading System' *User Modeling and User-Adapted Interaction* Vol. 4 pp 1-19, Kluwer Academic Publishers.

Bratko I. (1991) *'PROLOG: Programming for Artificial Intelligence'* Addison-Wesley Publishing.

Bratko I. (1993) 'Machine Learning in artificial intelligence' *Artificial Intelligence in Engineering*, Vol. 8 No. 3 pp 159-164.

Brown J. S. and Burton R. R. (1978) 'Diagnostic models for procedural bugs in basic mathematical skills' *Cognitive Science*, Vol. 2 pp 155-92.

Brown J. S. and Burton R. R. (1978b) 'A paradigmatic example of an artificially intelligent instructional system' *Int. Journal. of Man-Machine Studies*, Vol. 10 pp 323-339.

Brown P.J. (1987) 'Turning ideas into products: the Guide system' *Hypertext'97 Proceedings*, North Carolina, pp 33-40, November 13-15.

Brusilovsky P. and Pesin L. (1994) 'ISIS-Tutor: An Intelligent Learning Environment' *Proceedings of JCKBSE'94 Japanese-CIS Symposium on knowledge-based software engineering.* Perelavl-Zalesski, May 10-13, Tokyo.

Brusilovsky P. (1994a) 'Adaptive Hypermedia: An attempt to analyse and generalise' *Proceedings of UM'94 Fourth International Conference on User Modelling.*

Brusilovsky P. (1994b) 'Student model centred architecture for intelligent learning environments' *UM'94 Fourth International Conference on User Modelling*, August 15-19 1994.

Brusilovsky P. (1996) 'Methods and techniques of adaptive hypermedia' *User modelling and User-Adapted Interaction.* Special issue on: Adaptive Hypertext and Hypermedia, Vol. 6 No. 2-3, July 1996.

Brusilovsky P. (1995) 'Intelligent Tutoring Systems for World-Wide Web' *Proceedings of Third International WWW Conference, WWW'95.* http://www.igd.fhg.de/www/www95/proceedings/posters/48

Brusilovsky P., Schwarz E. and Weber G. (1996a) 'ELM-ART: an Intelligent Tutoring System on World Wide Web' *Proceedings of ITS'96.*

Brusilovsky P., Schwarz E. and Weber G. (1996b) 'A Tool for Developing Adaptive Electronic Textbooks on WWW' *Proceedings of WebNet'96, World Conference on the Web Society*, pp 64-69, Charlottesville, AACE.

Brusilovsky P., Ritter S. and Schwarz E. (1997) 'Distributed intelligent tutoring on the Web' *Proceedings of the 8th World Conference of the AIED Society*, Kobe, Japan, August 18-22 1997.

Brusilovsky P. and Eklund J (1999) 'A Study of User Model Based Link Annotation in Educational Hypermedia' http://www.iicm.edu/jucs_4_4/a_study_of_user/paper.html

Burns H.L. and Capps C.G. (1988) 'Foundations of Intelligent Tutoring Systems: An Introduction' Polson M.C. and Richardson J.J. (eds) *Foundations of Intelligent Tutoring Systems* pp 1-19.

Burton R.R. and Brown J.S. (1982) 'An investigation of computer coaching for informal learning activities' Sleeman D. and Brown J. (eds) *Intelligent Tutoring Systems*, Orlando, FL: Academic Press.

Bush V. (1945) 'As We May Think' *The Atlantic Monthly*, July 1945.

Campbell B. and Goodman J. M. (1988) 'HAM: A general purpose hypertext abstract machine' *Communications of the ACM* 31, 7(July) pp 856-861.

Carbonell J.G. (1983) 'Derivational analogy in problem solving and knowledge acquisition' *Proc. Int. Machine Learning Workshop,* Illinois.

Carr B. and Goldstein (1977) 'Overlays: A Theory of Modelling for Computer Aided Instruction' *AI Memo 406*, MIT, Cambridge, MA.

Chan T.W. (1989) 'Learning Companion Systems' *Ph.D. thesis*, Department of Computer Science, University of Illinois, Urbana-Champain.

Chan T.W. and Baskin A.B. (1990) 'Learning Companion Systems' in Frasson C and Gauthier G. (eds).. *Intelligent Tutoring Systems: At the Crossroads of Artificial Intelligence and Education*, Ablex, Norwood.

Cheeseman P. and Self M., Kelly J., Taylor W. Freeman D. and Stutz J. (1988) 'Bayesian classification' *Proceedings of AAAI'88.*

Clancey W. J. (1983) 'GUIDON' *Journal of Computer-Based Instruction*, Vol. 10 No. 1 pp 8-14.

Clark P. and Niblett T. (1989) 'The CN2 induction Algorithm' *Machine Learning*, Vol. 3 pp 261-283.

Cohen P.R. and Feigenbaum E.A. (1982) *The Handbook of Artificial Intelligence*, Cohen P.R.and Feigenbaum E.A. (Eds).

Conklin J. (1987) 'Hypertext: An Introduction and Survey' *IEEE Computer* 2(9) pp 17-41, September 1987.

Costa E., Duchenoy S. and Kodratuff Y. (1988) 'A resolution based method for discovering students' misconceptions' in Self J. (eds) *Artificial Intelligence and Human Learning: Intelligent Computer-Aided Instruction* pp 156-164.

Crabtree B. I., Soltysiak S. J. (1998) 'Identifying and tracking changing interests' *International Journal on Digital Libraries* 1998.

DeJong, G. and Mooney R. (1986) 'Explanation-based learning: An alternative view' *Machine Learning*, Vol. 1 pp 145-176.

de La Passardiere B., A. Dufresne (1992) 'Adaptive Navigational Tools for Educational Hypermedia' in Tomek I. (eds) *Computer Assisted Learning*, pp 555-567, Berlin, New York: Springer-Verlag.

Dwyer D, Barbieri K. and Doerr H. (1999) 'Creating a Virtual Classroom for Interactive Education on the Web' *WWW'95 Proceedings.* http://www.igd.fhg.de/www/www95/proceedings/papers/62/ctc.virtu.../ctc.virtual.class.htm

Easterlin J.D. and Langley P. (1985) 'A frame work for concept formation' *Proceedings of 7th Annual. Conference of Cognitive Science Society*, Irvine, CA.

Edwards D. and Hardman M.L. (1989) ''Lost in Hyperspace': Cognitive Mapping and Navigation in a Hypertext Environment'' in McAleese R. (eds) *.Hypertext: theory into practice*, pp 105-125.

Edwards P., Bayer D., Green C.L. and Payne T. R. (1996) 'Experience with learning agents which Manage Internet-Based Information' *In AAAI Spring Symposium on Machine Learning in Information Access*, pp 31-40, Menlo Park, CA-AAAAI.

Edwards P., Green C.L., Lockier P.C. and Lukins T.C. (1997) 'Exploiting Learning Technologies for World Wide Web Agents' *Intelligent World Wide Web Agents*. Colloquium organised by Professional Group C4 (Artificial Intelligence). Monday, 17 March 1997, Savoy Place, London.

Eklund J., Brusilovsky P. and Schwarz E. (1997) 'Adaptive Textbooks on the World Wide Web' *AusWeb97 Third Australian World Wide Web Conference.*

Elm W. C. and Woods D. D. (1985) 'Getting lost: A case study in interface design' *Proceedings of the Human Factors Society*, pp 927-931.

Elsom-Cook M. (1988) 'Guided discovery tutoring and bounded user modelling' in Self J. (eds) *Artificial Intelligence and Human Learning: Intelligent Computer-Aided Instructions.*

Elsom-Cook M. (1993) 'Student modelling in intelligent tutoring systems' *Artificial Intelligence Review* Vol. 7 pp 227-240.

Fayyad U. M., Djorgovski S.G. and Weir N. (1996) 'From Digitized Images to Online Catalogs: Data Mining a Sky Survey' *AI magazine* Vol. 17 No. 2, Summer 1996.

Fisher, D. (1987) 'Knowledge acquisition via incremental conceptual clustering' *Machine Learning*, 2, pp 139-172.

Furnas, G. W. (1986) 'Generalised fisheye views' *CHI'86 Proceedings of Human Factors in Computing Systems*, pp 16-23 ACM Boston April 13-9 New York 1986.

Furner J. *et al* (1999) 'Evaluation of Hypertext/Hypermedia Information Retrieval' *Proceedings of the Second Mira Workshop.* http://www.dcs.gla.ac.uk/mira/workshops/padua_procs/htir.html

Gilmore D.J. (1986). Concept Learning: alternative methods of focussing. *Proc. International Meeting on Advances in Learning*, Les Arcs, France.

Gilmore D. and Self J. (1988) 'The application of machine learning to intelligent tutoring systems' in Self J. (eds) *Artificial Intelligence and Human Learning: Intelligent Computer-Aided Instructions,*

Goodman D. (1988) *The Complete HyperCard Handbook*, Second Edition, Bantam Books, October 1988.

Gonschorek M. and Herzog C.H. (1995) 'Using Hypertext for an Adaptive Help System in an Intelligent Tutoring System' Jim Greer (Eds), *Proceedings of AI-ED'95, World Conference on Artificial Intelligence in Education*, pp 274-281, Washington, DC; August 16-19.

Halasz F.G. (1988) 'A Multimedia Idea Processing Environment' *Interactive Multimedia.* Edited by Ambron, Sueann and Hooper, Kristina, Microsoft Press, 1988.

Hall R. (1989) 'Computational approaches to analogical reasoning' *Artificial Intelligence*, 39(1).

Hinton G.E. and Sejnowski J.J. (1986) 'Learning and relearning in Boltzmann Machines' *Parallel Distributed Processing*, Rumelhart D.E., McClelland J.L. (eds), the PDP Research Group, pp 282-317. Cambridge, MA: MIT Press.

Hirashima T., Matsuda N., Nomoto T. and Toyoda J. (1998). 'Context-Sensitive Filtering for Browsing in Hypertext' *Proceedings of IUI'98*, San Francisco CA, USA.

Hohl H., Bocker H. D. and Gunzenhauser R. (1996) 'Hypadapter: An Adaptive Hypertext System for Exploratory Learning and Programming' *User-Adapted Interaction An International Journal*, Vol. 6 No. 2-3, July 1996.

Holt P., Dubs S., Jones M. and Greer J. (1991) 'The State of Student Modelling. *Student Modelling: The Key to Individualised Knowledge-Based Instruction*' Greer J.E. and McCalla G.I. (eds), Nato ASI Series.

Hook K., Karlgren J., Waern A., Dahlback N., Jansson C.G., Karlgren K. and Lemaire B. (1996) 'A Glass Box Approach to Adaptive Hypermedia' *User Modeling and User-Adapted Interaction*, 6.

Hook K. (1997) 'Evaluating the Utility and Usability of an Adaptive Hypermedia System' IUI'97 Proceedings, pp 179-186, Orlando, Florida USA, http://www.sics.se/~kia/

Hopfield J.J. (1982) 'Neural networks and physical systems with emergent collective computational abilities'. *Proceedings of the National Academy of Sciences USA*, 79(8), pp 2554-2558.

Hutchinson A. (1994) *Algorithmic Learning*. Oxford University Press Inc, New York.

Kader-Cabelli S.T. (1988) Analogy-from a unified perspective. In *Helman*.

Kaplan C, Fenwick J. and Chen J. (1993) 'Adaptive Hypertext Navigation Based On User Goals and Context' *User Modelling and User-Adapted Interaction*, Vol. 3 pp 193-220.

Kass R. (1989) 'Student Modelling in Intelligent Tutoring Systems – Implications for User Modelling' in Kobsa A. and Wahlster W. (eds) *User Models in Dialog Systems*, pp 386-410, Springer-Verlag.

Kobsa A. (1990) 'Modelling the User's Conceptual Knowledge in BGP-MS, a User Modelling Shell System' *Computational Intelligence*, Vol. 6 pp 193-208.

Kobsa A. and Wahlster W. (1989) *User Models in Dialog Systems* Kobsa A. and Wahlster W. (eds), Springer-Verlag.

Kobsa A., Müller D., and Nill A. (1994) 'KN-AHS: An Adaptive Hypertext Client of the User Modelling System BGP-MS' *Proceedings of the Fourth International Conference on User Modelling*, Hyannis, MA, pp 99-105.

Koedinger K. R., Anderson J. R., Hadley W. H. and Mark M. A. (1995) 'Intelligent tutoring goes to school in the big city' *International Journal of Artificial Intelligence in Education* Vol 8.

Kohonen T. (1984) *Self Organization and Associative Memory*. Berlin: Springer-Verlag.

Laog K. (1995) "NewsWeeder:Learning to filter Netnews", *Proc. 12th International Machine Learning Conference* (ML95), Morgan Kaufmann, San Francisco, pp 331-339.

LangleyP., Ohlsson S. and Sage S. (1984) A machine learning approach to student modelling. *Report CMU-RI-TR-84-7*, The Robotics Institute, Carnegie - Mellon University

Laurillard D., Preece J., Shneiderman B., Neal L. and Wærn Y. (1998) 'Distance Learning: Is It the End of Education as Most of Us Know it?' *CHI'98 Proceedings*, 18-23 April 1998.

Lieberman H. (1995) 'Letizia: An Agent That Assists Web Browsing' *Proceedings of 14th International Joint Conference on Artificial Intelligence, Montreal, Canada.*

Littman M. (1991) 'MaxiBook: The SuperBook™ System educates a dumb terminal' *Bellcore Technical Memorandum*, Bellcore, Morristown, New Jersey.

Luger, G. F. and W. A. Stubhlefield (1993) *'Artificial Intelligence'* (The Benjamin Cummings Publishing).

Maes P. (1994) 'Agents that Reduce Work and Information Overload' *Communications of the ACM*, July 1994.

Mathe N. and Chen J. (1994) 'A User-Centred Approach to Adaptive Hypertext based on an Information Relevance Model' *UM'94 Fourth International Conference on User Modelling.*

McCalla G.I. and Greer J.E. (1991) 'Granularity-Based Reasoning and Belief Revision in Student Models' *The Key to Individualised Knowledge-Based Instruction*, Greer J.E. and McCalla G. I. (eds) Nato ASI Series.

McCarthy, J. (1958) 'Programs with common sense' *Proceedings of the symposium on Mechanisation of Thought Processes*, Vol. 1, pp 77-84, London.

McEneaney J. E.(1999) 'Visualizing and Assesing Navigation in Hypertext' *Proceedings of Hypertext '99*, Darmstadth, Germany.

Michalski R.S. (1980) 'Pattern recognition as rule-guided inductive inference' *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-2*, pp 349-361.

Michalski R.S. (1986) 'Understanding the nature of learning: Issues and Research Directions' *Machine Learning, An Artificial Intelligence Approach* .Michalski R.S., .Carbonell J.G. and T. M. Mitchell (eds) Vol.2.

Michell T.M. (1997) 'Does Machine Learning Really Work?' *AI magozine*, Vol. 18, No.3.

Mitchell T.M., Keller R.M. and Kedar-Cabelli S.T. (1986) 'Explanation-based generalization: A unifying view' *Mochine Learning*, Vol.1, pp 47-80.

Moulinier I. (1996) 'A Framework for comparing text categorisation approaches' *AAAI Spring Symposium on Machine Learning and Information Access.*

Nakabayashi K., Maruyama M., Koike Y., Kato Y., Touhei H. and Fukuhara Y. (1997) 'Architecture of an Intelligent Tutoring system on the WWW' *Proceedings of the 8th World Conference of the AIED Society*, Kobe, Japan, August 18-22 1997.

Nelson T. (1965) 'A File Structure for the Complex, The Changing and Indeterminate' *ACM 20th Notional Conference.*

Nelson T. (1987) 'All for One for All' *Hypertext'87 Proceedings*, November 1987.

Nielsen J. (1990) 'The art of Navigating through Hypertext' *Communications of ACM*, Vol.33 No.3, March 1990.

Nielsen J. (1995a) 'The Architecture of Hypertext Systems' *Multimedia and Hypertext, The Internet and Beyond*, pp 131-157.

Nielsen J. (1995b) 'Defining . Hypertext, Hypermedia, and Multimedia' *Multimedia and Hypertext: The Internet and Beyond* pp 1-17.

Newell A. and H.A. Simon (1961) 'GPS, a program that simulates human thought' in Billing H. and Oldenbourg R. (eds), *Lernende Automaten*, pp 109-124, Munich, Germany.

Ohlsson S. (1987) 'Some Principles of Intelligent Tutoring' in Lawler R.W and Yazdani M. (eds) *Artificial Intelligence and Education; Learning environments and tutoring systems* pp 203-237.

O'Keefe R.A. (1983). 'Concept formation from very large training sets' *In Proceedings of the 8th International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany: Morgan Kaufmann.

Or-Bach R. and Bar-On E. (1989) 'PROBIT- Developing a Diagnostic intelligent tutor' in Bierman D., Breuker J. and Sandberg J (eds), *Artificial Intelligence and Education. Proceedings of the 4th International Conference on AI and Education*, 24-26 May 1989, Amsterdam, Netherlands.

Orwant J. (1995) 'Heterogeneous Learning in the Doppelgänger User Modeling System' *User Modeling and User-Adapted Interaction*, 4: 107-130.

O'Shea T. and Self J. (1983) 'A history of computing in education' in O'Shea T. and Self J. (eds) *Learning and teaching with computers: Artificial Intelligence in Education*, Prentice-Hall.

Pazzani M., Muramatsu J. and Billsus D. (1997). 'Syskill & Webert: Identifying interesting web sites' http://128.195.1.46:80/~pazzani/RTF/AAA1.html.

Pazzani M. and Billsus D. (1997) 'Learning and revising user profiles: the identification of interesting web sites' *Machine learning*, 27(3), 1997.

Payne T.R and Edwards P. (1997) 'Interface agents that Learn: An investigation Of Learning Issues in a Mail Agent Interface' *Applied Artificial Intelligence*, 11(1) pp 1-32.

Perez T.A., Gutierrez J. and Lopsteguy P. (1995) 'An Adaptive Hypermedia System' *Proceedings of AI-ED'95 World Conference on Artificial Intelligence in Education*, Greer J. (Eds), pp 351-358, Washington, DC; August 16-19 1995.

Quinlan I.R. (1983) 'Learning efficient classification procedures and their application to chess end-games' in Michalski R.S., Carbonell J.G. and Mitchell T.M. (eds), *Machine Learning: An Artificial Intelligence Approach*Morgan Kaufmann, pp 463-482.

Quinlan I.R. (1986) 'Induction of Decision Trees' *Machine Learning*, Vol.1, pp 181-106.

Quinlan I. R. (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

Rada R. and Murphy C. (1992) 'Searching versus browsing in hypertext' *Hypermedia*, Vol.4 No.1.

Rich E. and Knight K. (1991) *Artificial Intelligence*. Mc Graw-Hill, Inc.

Ritter S. (1997) 'Pat Online: A Model-tracing tutor on the World-wide Web' *Proceedings of the workshop 'Intelligent Educational Systems on the World Wide Web', 8th World Conference of the AIED Society*, Kobe, Japan, pp 18-22 August 1997.

Schlimmer J. C. and Fisher D. (1986) 'A case study of incremental concept induction' *Proceedings of the Fifth Nationol Conference on Artificial Intelligence*, pp 496-501, Morgan Kaufmann.

Schwarz E., Brusilovsky P. and Weber G. (1996) 'World-Wide Intelligent *Textbooks' ED-MEDIA '96 Proceedings.*

Self J.A. (1987) 'The application of machine learning to student modelling' in Lawler R.W. and Yazdani M. (eds), *Artificiol Intelligence ond Educotion* Vol.1 pp 267-280.

Self J.A. (1988) 'The use of belief system for student modelling' *Proceedings of 1st European Congress on Artificial Intelligence and Troining*, Lille.

Self J. A. (1990) 'By passing the intractable problem of student modelling' in Frasson C. and Gauthier G. (eds) *Intelligent Tutoring Systems: At the Crossroads of Artificial Intelligence and Education*, Ablex, Norwood.

Shannon C.E. (1950) Programming a computer for playing chess. *Philosophical Magazine*, 41(4), pp 256-275.

Sheth B. and Maes P. (1993) 'Evolving Agents for Personalised Information Filtering' *IEEE Conference on Artificial Intelligence for Applications.*

Shneiderman and Kearsley (1989) 'Hypertext Hands-On!' in Shneiderman B. and Kearsley G. (eds) *An. Inrtoduction :of a New way of Organising and Accessing Information*. Addision-Wesley, USA 1989.

Shortliffe E.H. (1976) 'Computer-based Medical Consultations: MYCIN' *American Elsevier Publishers*, NewYork.

Signore O. (1995) 'Issues on Hypertext design' *6th International DEXA Conference and Workshop on Database and Expert System Applications, DEXA '95*, September 4-8, 1995, London.

Simon H. (1983) 'Why Should Machine Learn?' Michalski R., Carbonell J. and Mitchell T.M. (eds), *Machine Learning: an Artificial Intelligence Approach*, Palo Alto:Tioga.

Sims M. H. (1987) 'Empirical and analytic discovery in IL' *Proceedings of the Fourth International Workshop on Machine Learning*. Los Altos, CA: Morgan Kaufmann.

Smith A.S., Blaodford A. and Torrance S (1996) 'Default parameter selection for multiple protein sequence alignment: An application of the ID3 learning *algorithm' Foculty of Technology Technical Report, TR7.96, ISSN 1 362-2285*, Middlesex University.

Sleeman D.H. (1982) 'Inferring (mal) Rules from Pupil's Protocols' *in Proceedings. of the 6th Europeon Conference on Artificial Intelligence*, Orsay, France, pp 160-164.

Sparck Jones K. (1989) 'Realism About User Modeling' in Kobsa A. and Wahlster W. (eds) *User Models in Dialog Systems*, pp 336-363, Springer-Verlag.

Stepp and Michalski (1986) 'Conceptual Clustering: Inventing Goal-Oriented Classifications of Structured Objects' in Michalski R.S., Carbonell J. G. and Mitchell T.M. (eds) *Machine Learning: An Artificial Intelligence Approach* Vol.2 Los Alto, CA: Morgan Kaufmann.

Stotts D. P. and Furuta R. (1991) 'Dynamic Adaptation of Hypertext Structure' *Hypertext'91 Proceedings*, pp 219-231, December 1991.

Sun C. T., Ching Y. T. and Lin F.X., (1995) 'Modelling Hypermedia Navigation: An AI Approach' *Proceedings of AI-ED'5*, Greer J. (Eds),Washington, DC; August 16-19, 1995, AACE.

Taylor and Self (1990) 'Monitoring hypertext users' *Interacting with Computers* Vol. 2 No. 3.

Theng Y. L. (1997) 'Addressing the 'lost in hypespace' problem in hypertext'. *PhD Thesis, School of Computing Science, Middlesex University*, September, 1997.

Thornton C. J. (1992) *Techniques in Computational Learning: An introduction*, pp 70-85.

Turing A.M., Strachey C., Bates M.A. and Bowden B.V. (1953) 'Digital computers applied to games' in Bowden B.V.(eds), *Foster than Thought*, pp 286-310, Pitman, London, Turing is believed to be sole author of the section of this paper that deals with chess.

UKOLN (1998) 'DC-dot'
URL: http: //www.ukoln.ac.uk/metadata/resources/dc.html

Utgoff, P. E. (1989) 'Incremental Induction of Decision Trees' *Machine Learning*, Vol. 4 pp 161-186.

Utting K. and Yankelovich N. (1989) 'Context and Orientation in Hypermedia Networks' *ACM Transactions on Information Systems* Vol. 7 No. 1 pp 58-84, January 1989.

VanLehn K. (1988) 'Student Modeling' in Polson M.C. and Richardson J.J. (eds), *Foundations of Intelligent Tutoring Systems*.

Young R.M., Plotkin G.D. and Linz R.F. (1977). 'Analysis of an extended concept learning task' *IJCAI-77 Proceedings*, Los Altos, CA: Morgan Kaufmann. Inc.

Walker J. (1987) 'Document Examiner: Delivery Interface for Hypertext Documents' *Hypertext'87 Proceedings*, November.

Weber. G. and Mollenberg A. (1994) 'ELM-PE: A knowledge-based programming environment for learning LISP' ED-MEDIA'94 Proceedings, Vancouver, Canada, pp 557-562.

Weber G. and Specht M. (1997) 'User modelling and Adaptive Navigation Support in WWW-Based Tutoring Systems' *UM'97 Proceedings of the 6th International conference on User Modelling*, Jameson A., Paris C. and Tasso C., pp 289-300.

Weizenbaum J. (1965) 'ELIZA- a computer program for the study natural language communication between man and machine' *Communications of the Association for Computing Machinery*, 9(1) pp 36-45.

Winston P. H. (1975) 'Learning structural descriptions from examples' in Winston P.H. (eds) *The Psychology of Computer Vision*, New York: McGraw-Hill.

Woodhead N. (1991) 'Hypertext and Hypermedia: Theory and Applications' Addision-Wesley.

Wolstencroft J. (1989) Restructuring, reminding and repair: What's missing from models of analogy. *AICOM*, 2(2), pp 58-71.

Woolf B. P. aod Murray T. (1991) 'Using Machine learning to Advice A Student Model' in Greer J. E. and McCalla G. I. (eds) *The key to Individualised Knowledge-Based Instruction* pp 127-146, Nato ASI Series 1991.

Acronyms and Abbreviations

AHS	Adaptive Hypermedia System
AI	Artificial Intelligence
ANS	Adaptive Navigation Support
ASCII	American Standard Code for Information Interchange
FTP	File Transfer Protocol
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
ICAI	Intelligent Computer Aided Instruction
ITS	Intelligent Tutoring System
MEMEX	Memory Extender
MLC	Machine Learning Component
URL	Uniform Resource Location
Web	World Wide Web
WWW	World Wide Web

Appendix A

Appendix A

The information contained in this appendix relates to the formation of the MLTutor attribute database, which is derived from the WWW documents loaded into the system. The attribute database formation principles are described in detail in chapter 6 and 7.

A.1 WWW documents in MLTutor

The prototype of MLTutor contains four Web sites. In total, across the four sites, there are 133 pages containing information on 'Environmental Science' issues. Each page is listed below indicating the keywords and phrases selected for the page.

WEBSITE 1

Page number page001
File name acid.buildings.htm
Page title buildings

	Keyword	Destination File name	Destination Page Title
Author's	lakes and rivers	acid.lakes.htm	lakes and rivers
	trees	acid.trees.htm	trees
	people	acid.people.htm	people
	but how big a problem is it?	Acid.how.big.problem.htm	how much trouble is acid rain?
	Return to air pollution page	air.pollution.htm	air pollution
	return to acid rain page	acid.home.htm	acid rain
Non Author's	buildings		
	acid rain		

Page number page002
File name acid.formation.htm
Page title how is acid rain formed?

	Keyword	Destination File name	Destination Page Title
Author's	so what's the problem?	acid.what.problem.htm	so what's the problem?
	where do these gases come from?	acid.where.from.htm	acid rain how is it made?
	return to air pollution page	air.pollution.htm	air pollution
	return to acid rain page	acid.home.htm	acid rain
Non Author's	how is acid rain formed?		
	carbondioxide		
	chlorine		
	carbonic acid		
	hydrochloric acid		
	sulphur dioxide		
	nitrogen		
	nitrogen oxides		

Page number page003
File name acid.home.htm
Page title acid rain

	Keyword	Destination File name	Destination Page Title
Author's	what is this acid rain stuff anyway?	acid.what.is.it.htm	what is acid rain?
	where does it come from?	acid.where.from.htm	acid rain how is it made?
	what's all the fuss about?	acid.what.problem.htm	so what's the problem?

	how big a problem is it?	acid.how.big.problem.htm	how much trouble is acid rain?
	what can we do about it?	acid.what.now.htm	what can we do?
	what's the government done?	acid.laws.htm	acid rain legislation
	return to air pollution page	air.pollution.htm	air pollution
Non Author's	acid rain		

Page number	page004
File name	acid.how.big.problem.htm
Page title	how much trouble is acid rain?

	Keyword	**Destination File name**	**Destination Page Title**
Author's	so what can we do?	acid.what.now.htm	what can we do?
	return to air pollution page	air.pollution.htm	air pollution
	return to acid rain page	acid.home.htm	acid rain
Non Author's	how much trouble is acid rain?		
	acid rain		
	global problem		
	sulphur dioxide		
	power station		

Page number	page005
File name	acid.lakes.htm
Page title	lakes and rivers

	Keyword	**Destination File name**	**Destination Page Title**
Author's	trees	acid.trees.htm	lakes and rivers
	people	acid.people.htm	trees
	buildings	acid.buildings.htm	people
	but how big a problem is it?	acid.how.big.problem.htm	how much trouble is acid rain?
	return to air pollution page	air.pollution.htm	air pollution
Non Author's	lakes and river		
	pH		
	acidic		
	aluminium		
	toxic metals		
	trees		
	people		
	buildings		
	acidic water		

Page number	page006
File name	acid.laws.htm
Page title	legislation

	Keyword	**Destination File name**	**Destination Page Title**
Author's	return to air pollution page	air.pollution.htm	air pollution
	return to acid rain page	acid.home.htm	acid rain
Non Author's	legislation		
	acid rain		
	global problem		
	air pollution		
	sulphur emission		
	nitrogen oxides		

Page number	page007
File name	acid.people.htm
Page title	people

	Keyword	**Destination File name**	**Destination Page Title**
Author's	trees	acid.trees.htm	trees
	lakes and rivers	acid.lakes.htm	lakes an rivers
	buildings	acid.buildings.htm	buildings
	but how big a problem is it?	acid.how.big.problem.htm	how much trouble is acid rain?
	return to air pollution page	air.pollution.htm	air pollution

	return to acid rain page	acid.home.htm	acid rain

Non Author's
people
acid rain
toxic metals
acidic
diarrhoea
damaged livers
kidneys

Page number page008
File name acid.trees.htm
Page title trees

	Keyword	Destination File name	Destination Page Title
Author's	buildings	acid.buildings.htm	buildings
	lakes and rivers	acid.lakes.htm	lakes and rivers
	people	acid.people.htm	people
	but how big a problem is it?	acid.how.big.problem.htm	how much trouble is acid rain?
	return to air pollution page	air.pollution.htm	air pollution
	return to acid rain page	acid.home.htm	acid rain

Non Author's
trees
acid rain
conifers
calcium
nutrients
magnesium

Page number page009
File name acid.what.is.it.htm
Page title what is acid rain?

	Keyword	Destination File name	Destination Page Title
Author's	form various acids	acid.formation.htm	how is acid rain formed?
	where does acid rain come from?	acid.where.from.htm	acid rain how is it made?
	return to air pollution page	air.pollution.htm	air pollution
	return to acid rain page	acid.home.htm	acid rain

Non Author's
what is acid rain?
acid rain
acidic
carbon dioxide
pH
chlorine
sulphuric acid

Page number page010
File name acid.what.now.htm
Page title what can we do?

	Keyword	Destination File name	Destination Page Title
Author's	greenhouse effect	greenhouse.effect.htm	the greenhouse effect
	laws	acid.laws.htm	legislation
	return to air pollution page	air.pollution.htm	air pollution
	return to acid rain page	acid.home.htm	acid rain

Non Author's
what can we do?
nitrogen oxides
sulphur dioxide
atmosphere
catalytic converter
carbon dioxide
sulphur emission
power station
fossil fuels

Appendix A

Page number	page011		
File name	acid.what.problem.htm		
Page title	so what's the problem?		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	buildings	acid.buildings.htm	buildings
	lakes and rivers	acid.lakes.htm	lakes and rivers
	trees	acid.trees.htm	trees
	people	acid.people.htm	people
	but how big a problem is it?	acid.how.big.problem.htm	how much trouble is acid rain?
	return to air pollution page	air.pollution.htm	air pollution
	return to acid rain page	acid.home.htm	acid rain
Non Author's	so what's the problem?		
	acid rain		

Page number	page012		
File name	acid.where.from.htm		
Page title	acid rain how is it made?		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	sulphuric acid	acid.formation.htm	how is acid rain formed?
	so what's the problem?	acid.what.problem.htm	so what's the problem?
	return to air pollution page	air.pollution.htm	air pollution
	return to acid rain page	acid.home.htm	acid rain
Non Author's	acid rain how is it made?		
	sulphur dioxide		
	nitrogen oxides		
	fossil fuels		
	water		

Page number	page 013		
File name	acir_rain.htm		
Page title	acid rain		
	Keyword	**Destination File name**	**Destination Page TItle**
Author's	tell me more about this acid rain	acid.home.htm	acid rain
Non Author's	acid rain		
	pH		
	acidic		
	acidity		
	sulphur dioxide		
	nitrogen oxides		
	hydrogen chloride		
	halons		
	chlorinated solvents		

Page number	page014		
File name	agrochem.htm		
Page title	agricultural chemicals		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	recalcitrant molecules	agrochem.recalcitrant.htm	recalcitrant molecules
	return to water pollution home page	water.htm	water pollution
Non Author's	agricultural chemicals		
	fertilisers		
	ecosystem		
	water		

Page number	page015		
File name	agrochem.ddt.htm		
Page title	ddt		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	return to recalcitrant molecules page	agrochem.recalcitrant.htm	recalcitrant molecules

Non Author's	ddt		
Page number	page016		
File name	agrochem.detergents.htm		
Page title	detergent formulation		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	return to recalcitrant molecules page	agrochem.recalcitrant.htm	recalcitrant molecules
Non Author's	detergent formulation		
	biodegradability		
	detergent		
	alkylbenzene sulphonate		
	las		
	abs		
Page number	page017		
File name	agrochem.recalcitrant.htm		
Page title	recalcitrant molecules		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	detergent formulation	agrochem.detergents.htm	detergent formulation
	return to agricultural chemical page	agrochem.htm	agricultural chemicals
	return to water home page	water.htm	water pollution
Non Author's	recalcitrant molecules		
	recalcitrant		
	polyethylene		
	hydrocarbons		
	ddt		
	water		
	detergent formulation		
Page number	page018		
File name	air.pollution.htm		
Page title	air pollution		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	acid rain	acid.home.htm	acid rain
	domestic smoke	domestic.smoke.htm	domestic smoke
	smog	smog.htm	smog
	the greenhouse effect	greenhouse.effect.htm	the greenhouse effect
	particulates	particulates.htm	particulates
	ozone layer depletion	oz_depl.htm	ozone layer depletion
Non Author's	air pollution		
Page number	page019		
File name	aral.sea.htm		
Page title	the aral sea		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	return to water pollution home page	water.htm	water pollution
Non Author's	the aral sea		
	water		
	lakes and rivers		
Page number	page020		
File name	case1_1.htm		
Page title	oil spill!		
	Keyword	**Destination File name**	**Destination Page Title**
Author's			
Non Author's	oil spill!		
	chemical dispersants		
	clean up ship		

Appendix A

Page number page021
File name case1_5.htm
Page title oil spill!

	Keyword	Destination File name	Destination Page Title
Author's	return to oil slicks home page	oil.slicks.htm	oil spills
	try this scenario again	case1_1.htm	oil spill!
	try another scenario	oil.you.clean.htm	now it's your turn!

Non Author's oil spill!

Page number page022
File name case2_1.htm
Page title oil slick!

	Keyword	Destination File name	Destination Page Title
Author's	send out a clean-up ship	case2_3.htm	oil slicks
	spray the slick with chemical dispersants	case2_2.htm	oil slicks
	take some photos of the slick	case2_6.htm	oil slicks

Non Author's oil slick!

Page number page023
File name case2_10.htm
Page title oil slick!

	Keyword	Destination File name	Destination Page Title
Author's	return to oil slicks home page	oil.slicks.htm	oil spills
	try this scenario again	case2_1.htm	oil slicks
	try another scenario	oil.you.clean.htm	now it's your turn!

Non Author's oil slick!
chemical dispersants

Page number page024
File name case2_11.htm
Page title oil slick!

	Keyword	Destination File name	Destination Page Title
Author's			

Non Author's oil slick!
chemical dispersants
send out a clean-up ship

Page number page025
File name case2_2.htm
Page title oil slick!

	Keyword	Destination File name	Destination Page Title
Author's	return to oil slicks home page	oil.slicks.htm	oil spills
	try this scenario again	case2_1.htm	oil slicks
	try another scenario	oil.you.clean.htm	now it's your turn!

Non Author's oil slick!
chemical dispersants

Page number page026
File name case2_3.htm
Page title oil slick!

	Keyword	Destination File name	Destination Page Title
Author's	continue with the clean-up operation	case2_4.htm	oil slicks
	spray the slick with chemical dispersants	case2_5.htm	oil slicks

Non Author's	oil slick!
	chemical dispersants
	send out a clean-up ship

Page number	page027
File name	case2_4.htm
Page title	oil slick!

	Keyword	Destination File name	Destination Page Title
Author's	return to oil slicks home page	oil.slicks.htm	oil spills
	try this scenario again	case2_1.htm	oil slicks
	try another scenario	oil.you.clean.htm	now it's your turn!

Non Author's	oil slick!

Page number	page028
File name	case2_6.htm
Page title	oil slick!

	Keyword	Destination File name	Destination Page Title
Author's	send out a clean-up ship	case2_7.htm	oil slicks
	spray the slick with chemical dispersants	case2_10.htm	oil slick!
	sell the photos to a newspaper	case2_11.htm	oil slick!

Non Author's	oil slick!

Page number	page029
File name	case2_7.htm
Page title	oil slick!

	Keyword	Destination File name	Destination Page Title
Author's	continue with the clean-up operation	case2_8.htm	oil slicks

Non Author's	oil slick!
	spray the slick with chemical dispersants
	send out a clean-up ship

Page number	page030
File name	case2_8.htm
Page title	oil slick!

	Keyword	Destination File name	Destination Page Title
Author's	return to oil slicks home page	oil.slicks.htm	oil spills
	try this scenario again	case2_1.htm	oil slicks
	try another scenario	oil.you.clean.htm	now it's your turn!

Non Author's	oil slick!

Page number	page031
File name	domestic.smoke.htm
Page title	domestic smoke

	Keyword	Destination File name	Destination Page Title
Author's	smog	smog.htm	smog
	return to air pollution page	air.pollution.htm	air pollution

Non Author's	smokeless fuel
	hydrocarbons
	smoke
	clean air acts

Page number	page032
File name	greenhouse.causes.htm
Page title	what causes the greenhouse effect?

	Keyword	Destination File name	Destination Page Title
Author's	acid rain	acid.home.htm	acid rain
	agriculture	agrochem.htm	agricultural chemicals
	waste disposal	wmanintro.htm	waste management and recycling
	what problems does it cause?	greenhouse.problems.htm	problems of greenhouse effect
	what can we do about it?	greenhouse.solutions.htm	the greenhouse effect possible solutions
	return to greenhouse effect page	greenhouse.effect.htm	the greenhouse effect
Non Author's	what causes the greenhouse effect?		
	greenhouse gases		
	water		
	nitrogen oxide		
	carbondioxide		
	methane		
	ozone		
	chlorofluorocarbon		
	cfc		
	refrigerant		
	solvent		
	aerosol propellant		
	infra-red radiation		
	fossil fuels		

Page number page033
File name greenhouse.effect.htm
Page title the greenhouse effect

	Keyword	Destination File name	Destination Page Title
Author's	what causes it?	greenhouse.causes.htm	what causes the greenhouse effect?
	what problems does it cause?	greenhouse.problems.htm	problems of the greenhouse effect
	what can we do?	greenhouse.solutions.htm	the greenhouse effect possible solutions
	what's the government done?	greenhouse.laws.htm	the greenhouse effect legislation
	return to air pollution page	air.pollution.htm	air pollution
	return to greenhouse effect page	greenhouse.effect.htm	the greenhouse effect
Non Author's	the greenhouse effect		
	atmosphere		
	infra-red radiation		
	greenhouse gases		

Page number page034
File name greenhouse.laws.htm
Page title the greenhouse effect legislation

	Keyword	Destination File name	Destination Page Title
Author's	return to greenhouse effect page	greenhouse.effect.htm	the greenhouse effect
Non Author's	the greenhouse effect legislation		
	global warming		
	greenhouse effect		
	cfc		
	greenhouse gases		

Page number page035
File name greenhouse.problems.htm
Page title problems of the greenhouse effect

	Keyword	Destination File name	Destination Page Title
Author's	return to greenhouse effect page	greenhouse.effect.htm	the greenhouse effect
	so what can we do?	greenhouse.solutions.htm	the greenhouse effect possible solutions
Non Author's	problems of the greenhouse effect		

greenhouse effect
ecosystems

Page number page036
File name greenhouse.solutions.htm
Page title the greenhouse effect possible solutions

	Keyword	Destination File name	Destination Page Title
Author's	acid rain	acid.home.htm	acid rain
	so what's the government done?	greenhouse.laws.htm	the greenhouse effect legislation
	return to greenhouse effect page	greenhouse.effect.htm	the greenhouse effect

Non Author's
the greenhouse effect possible solutions
cfc
carbondioxide
fossil tuels

Page number page037
File name LEMS.htm
Page title electromagnetic fields

	Keyword	Destination File name	Destination Page Title
Author's	return to environmental index	enviro_index.htm	
	return to light pollution	Light.htm	light pollution

Non Author's
electromagnetic fields
power lines
emission

Page number page038
File name Light.htm
Page title light pollution

	Keyword	Destination File name	Destination Page Title
Author's	the engineering solution to light pollution	LSol.htm	light pollution solutions
	ultraviolet light and the ozone layer	LSun.htm	light pollution
	other electromagnetic pollutions	LEMS.htm	electromagnetic fields
	return to environmental index	enviro_index.htm	

Non Author's
light pollution
pollutants
electromagnetic spectra

Page number page039
File name LSol.htm
Page title light pollution solutions

	Keyword	Destination File name	Destination Page Title
Author's	return to light pollution	Light.htm	light pollution
	return to environmental index	enviro_index.htm	

Non Author's
light pollution solutions
pollutants

Page number page040
File name LSun.htm
Page title ultra violet light

	Keyword	Destination File name	Destination Page Title
Author's	look at some of the things that reduce the ozone layer	air.pollution.htm	air pollution
	back to the light pollution index	Light.htm	light pollution
	other electromagnetic pollutants	LEMS.htm	electromagnetic fields
	return to environmental index	enviro_index.htm	

Non Author's	ultra violet light
	pollutants
	skin cancer

Page number	page041
File name	north.sea
Page title	the north sea

	Keyword	**Destination File name**	**Destination Page Title**
Author's	oil spills from tanker accidents	oil.slicks.htm	oil spills
	oil from other sources	oil.other.sources.htm	other sources of oil pollution
	agricultural chemicals	agrochem.htm	agricultural chemicals
	thermal pollution	thermal.htm	thermal pollution
	return to water pollution home page	water.htm	water pollution

Non Author's	the north sea
	water pollution
	pollutants

Page number	page042
File name	oil.braer.htm
Page title	the braer oil spill

	Keyword	**Destination File name**	**Destination Page Title**
Author's	chemical dispersants	oil.chemical.dispersant.htm	chemical dispersion
	booms and skimmers	oil.collect.htm	collection
	return to oil slicks home page	oil.slicks.htm	oil spills
	look at other oil spills	oil.other.spills.htm	other oil spills

Non Author's	the braer oil spill
Page number	page043
File name	oil.chemical.dispersant.htm
Page title	chemical dispersion

	Keyword	**Destination File name**	**Destination Page Title**
Author's	other clean-up methods	oil.cleanup.htm	how do we clean up oil slicks?
	return to oil slicks home page	oil.slicks.htm	oil spills

Non Author's	chemical dispersion
	oil slicks
	water
	pollutants

Page number	page044
File name	oil.cleanup.htm
Page title	how do we clean up oil slicks?

	Keyword	**Destination File name**	**Destination Page Title**
Author's	chemical dispersion	oil.chemical.dispersant.htm	chemical dispersion
	natural dispersion	oil.natural.dispersion.htm	natural dispersion
	collection	oil.collect.htm	collection
	return to oil slicks home page	oil.slicks.htm	oil spills

Non Author's	how do we clean up oil slicks?
	oil slicks

Page number	page045
File name	oil.collect.htm
Page title	collection

	Keyword	**Destination File name**	**Destination Page Title**
Author's	other clean-up methods	oil.cleanup.htm	how do we clean up oil slicks?
	return to oil slicks home page	oil.slicks.htm	oil spills
	chemical dispersion	oil.chemical.dispersant.htm	chemical dispersant
	gulf war	oil.gulf.war.htm	the gulf war
	natural dispersion	oil.natural.dispersion.htm	natural dispersion

Non Author's	collection

skimmer
solvents
water
oil spills

Page number page046
File name oil.exxon.htm
Page title exxon valdez

	Keyword	Destination File name	Destination Page Title
Author's	valdez principles	oil.exxon2.htm	the valdez principles
	return to oil slicks home page	oil.slicks.htm	oil spills
	look at some other spills	oil.other.spills.htm	other oil spills
Non Author's	exxon valdez		

Page number page047
File name oil.exxon2.htm
Page title the valdez principles

	Keyword	Destination File name	Destination Page Title
Author's	greenhouse effect	greenhouse.effect.htm	the greenhouse effect
	acid rain	acid.home.htm	acid rain
	smog	smog.htm	smog
	recycle materials	wmanintro.htm	waste management and recycling
	return to exxon valdez page	oil.exxon.htm	exxon valdez
Non Author's	the valdez principles		
	ozone layer		
	waste		
	pollutant		
	water		

Page number page048
File name oil.gulf.war.htm
Page title the gulf war

	Keyword	Destination File name	Destination Page Title
Author's	release of oil	oil.gulf.war.htm	the gulf war
	burning of kuwait's oil wells	oil.gulf.war.htm	the gulf war
	exxon valdez	oil.exxon.htm	exxon valdez
	braer	oil.braer.htm	the braer oil spill
	acid rain	acid.home.htm	acid rain
	return to oil slicks home page	oil.slicks.htm	oil spills
Non Author's	the gulf war		
	oil slick		
	acid rain		
	water		
	sulphur		
	skin cancer		
	oil spill		

Page number page049
File name oil.kuwait.htm
Page title kuwait burning

	Keyword	Destination File name	Destination Page Title
Author's	return to gulf war page	oil.gulf.war.htm	the gulf war
Non Author's	kuwait burning		

Page number page050
File name oil.measures.htm
Page title prevention of oil slicks

	Keyword	Destination File name	Destination Page Title

Author's	stop them having accidents	oil.measures.htm	prevention of oil slicks
	stop oil leaking	oil.measures.htm	prevention of oil slicks
	exxon valdez	oil.exxon.htm	exxon valdez
	return to oil slicks home page	oil.slicks.htm	oil spills
Non Author's	prevention of oil slicks		
	oil spills		

Page number	page051
File name	oil.natural.dispersant.htm
Page title	natural dispersion

	Keyword	Destination File name	Destination Page Title
Author's	braer	oil.braer.htm	the braer oil spill
	exxon valdez	oil.exxon.htm	exxon valdez
	return to oil slicks home page	oil.slicks.htm	oil spills
	other clean-up methods	oil.cleanup.htm	how do we clean up oil slicks?
Non Author's	natural dispersion		
	oil slicks		

Page number	page052
File name	oil.other.sources.htm
Page title	other sources of oil pollution

	Keyword	Destination File name	Destination Page Title
Author's	return to north sea pollution page	northsea.htm	the north sea
Non Author's	other sources of oil pollution		

Page number	page053
File name	oil.other.spills.htm
Page title	other oil spills

	Keyword	Destination File name	Destination Page Title
Author's	return to oil slicks home page	oil.slicks.htm	oil spills
	chemical dispersants	oil.chemical.dispersants.htm	chemical dispersants
Non Author's	other oil spills		

Page number	page054
File name	oil.slicks.htm
Page title	oil spills

	Keyword	Destination File name	Destination Page Title
Author's	return to water pollution home page	water.htm	water pollution
	exxon valdez	oil.exxon.htm	exxon valdez
	braer	oil.braer.htm	the braer oil spill
	the gulf war	oil.gulf.war.htm	the gulf war
	measures	oil.measures.htm	prevention of oil slicks
	techniques	oil.cleanup.htm	how do we clean up oil slicks?
	many other sources	oil.sources.htm	sources of oil pollution
	disaster management	oil.you.clean.htm	now it's your turn!
Non Author's	oil spills		
	oil slicks		

Page number	page055
File name	oil.sources.htm
Page title	sources of oil pollution

	Keyword	Destination File name	Destination Page Title
Author's	return to oil slicks home page	oil.slicks.htm	oil spills
Non Author's	sources of oil pollution		
	oil spills		

oil slicks

Page number page056
File name oil.you.clean.htm
Page title now it's your turn

	Keyword	Destination File name	Destination Page Title
Author's	return to oil slicks home page	oil.slicks.htm	oil spills
	clean up after a tanker runs around aground off shetland	case1_1.htm	oil spill!
	clean up an oil slick in a war zone	case2_1.htm	oil slick!

Non Author's now it's your turn
oil spills

Page number page057
File name organic2.htm
Page title organic wastes

	Keyword	Destination File name	Destination Page Title
Author's			

Non Author's organic wastes
composting
waste
particles
organic waste
nutrient
carbon
nitrogen
pH
potassium
carbon dioxide
biodegradable

Page number page058
File name organic3.htm
Page title organic wastes

	Keyword	Destination File name	Destination Page Title
Author's			

Non Author's organic wastes
anaerobic digestion
methane

Page number page059
File name oz_cfcs.htm
Page title chlorofluorocarbons

	Keyword	Destination File name	Destination Page Title
Author's	stratosphere	oz_parts.htm	the layer of the atmosphere
	troposphere	oz_parts.htm	the layer of the atmosphere
	threat to the ozone layer	oz_chem.htm	cfcs in the stratosphere
	destroy ozone molecules	oz_chem.htm	cfcs in the stratosphere
	global warming	greenhouse.effect.htm	the greenhouse effect

Non Author's chlorofluorocarbons
cfc
refrigerants
propellants

Page number page060
File name oz_chem.htm
Page title cfcs in the stratosphere

	Keyword	Destination File name	Destination Page Title

Author's

Non Author's cfcs in the stratosphere
stratosphere
troposphere
uv radiation
chlorine
ozone gas
catalytic
ozone layer
oxygen

Page number page061
File name oz_depl.htm
Page title ozone layer depletion

	Keyword	Destination File name	Destination Page Title
Author's	stratosphere	oz_parts.htm	the layer of the atmosphere

Non Author's ozone layer depletion
chlorofluorocarbons
cfc
ultra violet radiation
uv light
ozone layer
antarctica

Page number page062
File name oz_func.htm
Page title the ozone layer

	Keyword	Destination File name	Destination Page Title
Author's	stratosphere	oz_parts.htm	the layer of the atmosphere
	atmosphere	oz_layer.htm	where is the ozone what is ozone
	ozone gas	oz_gases.htm	what is ozone
	global warming	greenhouse.effect	the greenhouse effect

Non Author's the ozone layer
ultra violet light
uv
uvb
carbon dioxide
ozone layer

Page number page063
File name oz_gases.htm
Page title what is ozone

	Keyword	Destination File name	Destination Page Title
Author's	stratosphere	oz_parts.htm	the layer of the atmosphere
	smog	smog.htm	smog
	global warming	greenhouse.effect.htm	the greenhouse effect

Non Author's what is ozone
ozone layer
ultra violet radiation
uv
greenhouse gas

Page number page064
File name oz_intro.htm
Page title the ozone layer

	Keyword	Destination File name	Destination Page Title
Author's	atmosphere	oz_layer.htm	where is the ozone layer?

	ozone gas	oz_gases	what is ozone
	cfc's	oz_cfcs.htm	cholorfluorocarbons
Non Author's	the ozone layer		

Page number	page065
File name	oz_layer.htm
Page title	where is the ozone layer?

	Keyword	Destination File name	Destination Page Title
Author's	or follow this link	oz_parts.htm	the layer of the atmosphere
	ozone layer	oz_func.htm	the ozone layer
Non Author's	where is the ozone layer?		
	stratosphere		
	troposphere		
	mesosphere		
	atmosphere		
	troposphere		

Page number	page066
File name	oz_parts.htm
Page title	the layer of the atmosphere

	Keyword	Destination File name	Destination Page Title
Author's	the troposphere	oz_parts.htm	the layer of the atmosphere
	the tropopause	oz_parts.htm	the layer of the atmosphere
	the stratosphere	oz_parts.htm	the layer of the atmosphere
	the stratopause	oz_parts.htm	the layer of the atmosphere
	the mesosphere	oz_parts.htm	the layer of the atmosphere
	ozone layer	oz_func.htm	the ozone layer
	back to previous page	oz_layer.htm	where is the ozone layer?
Non Author's	the layer of the atmosphere		
	atmosphere		

Page number	page067
File name	oz_reg.htm
Page title	what has been done about cfcs?

	Keyword	Destination File name	Destination Page Title
Author's			
Non Author's	what has been done about cfcs?		
	ozone layer		
	montreal protocol		
	cfc's		
	copenhagen amendment		
	hcfc		
	ozone depletion		
	chlorine		
	stratosphere		

Page number	page068
File name	particulates.htm
Page title	particulates

	Keyword	Destination File name	Destination Page Title
Author's	acid rain	acid.buildings.htm	buildings
	return to air pollution page	air.pollution.htm	air pollution
Non Author's	particulates		
	atmosphere		
	particles		
	acid rain		
	sulphur dioxide		

Appendix A

Page number page069
File name radionuclides.htm
Page title radionuclides

	Keyword	Destination File name	Destination Page Title
Author's	return to air pollution page	air.pollution.htm	air pollution
Non Author's	radionuclides		
	atmosphere		
	particles		
	ions		
	x-rays		
	dna		
	cancer		
	ionisation		
	chernobyl		

Page number page070
File name rubbish.htm
Page title you have now entered the rubbish pages!

	Keyword	Destination File name	Destination Page Title
Author's	how much waste we produce	wmanintro.htm	waste management and recycling
	reusing waste	wmanreuse.htm	waste management and recycling
	how we can reduce the amount of waste we produce	wmanreduc.htm	waste management and recycling
	different sorts of waste	wmandust.htm	waste management and recycling
	aluminium drink cans	wmanalurecyc.htm	waste management and recycling
	paper	wmanpaperecyc.htm	waste management and recycling
	waste into compost	wmancompost.htm	waste management and recycling
	fuel incineration	wmanincin.htm	waste management and recycling
	turning rubbish into fuel	wmanfuel.htm	waste management and recycling
	composting	organic2.htm	organic wastes
	anaerobic digestion	organic3.htm	organic wastes
Non Author's	you have now entered the rubbish pages!		

Page number page071
File name smog.htm
Page title smog

	Keyword	Destination File name	Destination Page Title
Author's	particulates	particulates.htm	particulates
	return to air pollution page	air.pollution.htm	air pollution
Non Author's	smog		
	hydrocarbons		
	hydrocarbon emission		
	nitrogen emission		
	catalytic converter		

Page number page072
File name thermal.htm
Page title thermal pollution

	Keyword	Destination File name	Destination Page Title
Author's	return to water pollution home page	water.htm	water pollution
Non Author's	thermal pollution		

power station
enzymes
microbes
water

Page number page073
File name water.htm
Page title water pollution

	Keyword	Destination File name	Destination Page Title
Author's	oil spills	oil.slicks.htm	oil spills
	many other sources	oil.sources.htm	sources of oil pollution
	aral sea	aral.sea.htm	the aral sea
	north sea	northsea.htm	the north sea
	agriculture	agrochem.htm	agricultural chemicals
	thermal pollution	thermal.htm	thermal pollution

Non Author's water pollution
water
waste

Page number page073
File name water.htm
Page title water pollution

	Keyword	Destination File name	Destination Page Title
Author's	oil spills	oil.slicks.htm	oil spills
	many other sources	oil.sources.htm	sources of oil pollution
	aral sea	aral.sea.htm	the aral sea
	north sea	northsea.htm	the north sea
	agriculture	agrochem.htm	agricultural chemicals
	thermal pollution	thermal.htm	thermal pollution

Non Author's water pollution
water
waste

Page number page074
File name wmanalurecyc.htm
Page title waste management and recycling

	Keyword	Destination File name	Destination Page Title
Author's			

Non Author's waste management and recycling
aluminium recycling
aluminium
drink cans
power station
recycling plant

Page number page075
File name wmancompost.htm
Page title waste management and recycling

	Keyword	Destination File name	Destination Page Title
Author's			

Non Author's waste management and recycling
biowaste
organic waste
methane
natural gas
artificial fertiliser
nutrient
soil conditioner
landfill site

Page number	page076		
File name	wmancompost.htm		
Page title	waste management and recycling		
	Keyword	**Destination File name**	**Destination Page Title**
Author's			
Non Author's	waste management and recycling		
	recyclable material		
Page number	page077		
File name	wmandust.htm		
Page title	waste management and recycling		
	Keyword	**Destination File name**	**Destination Page Title**
Author's			
Non Author's	waste management and recycling		
	dustbin		
	rubbish		
Page number	page078		
File name	wmanfuel.htm		
Page title	waste management and recycling		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	click here	engin_index.htm	
Non Author's	waste management and recycling		
	dustbin		
	waste		
	the byker		
	pellets		
	recyclable material		
Page number	page079		
File name	wmanincin.htm		
Page title	waste management and recycling		
	Keyword	**Destination File name**	**Destination Page Title**
Author's			
Non Author's	waste management and recycling		
	waste		
	air pollution		
	incinerator		
	acid rain		
	exhaust gases		
	rubbish		
Page number	page080		
File name	wmanintro.htm		
Page title	waste management and recycling		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	click here	wmandust.htm	waste management and recycling
	reduce the amount that people throw away	wmanreduc.htm	waste management and recycling
	reuse rubbish	wmanreuse.htm	waste management and recycling
Non Author's	waste management and recycling		
	waste		
	air pollution		
	incinerator		
	acid rain		
	exhaust gases		
	rubbish		

Appendix A

Page number	page081		
File name	wmanirspec.htm		
Page title	waste management and recycling		
	Keyword	**Destination File name**	**Destination Page Title**
Author's			
Non Author's	waste management and recycling		
	plastic		
	infra red spectroscopy		
	plastic recycling		
Page number	page082		
File name	wmanpack.htm		
Page title	waste management and recycling		
	Keyword	**Destination File name**	**Destination Page Title**
Author's			
Non Author's	waste management and recycling		
	waste		
	infra red spectroscopy		
	plastic recycling		
	recyclable material		
	biodegrade		
Page number	page083		
File name	wmanpaperecyc.htm		
Page title	waste management and recycling		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	global warming	greenhouse.effect.htm	the greenhouse effect
Non Author's	waste management and recycling		
	paper		
	trees		
	paper recycling		
	recyclable material		
	greenhouse gas		
	carbon dioxide		
Page number	page084		
File name	wmanplasticsep.htm		
Page title	waste management and recycling		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	designing recyclable products	wmandesign.htm	recyclable material
	click here	wmanplastp.htm	waste management and recycling
	click here	wmanirspec.htm	waste management and recycling
Non Author's	waste management and recycling		
	types of plastics		
	recyclable material		
Page number	page085		
File name	wmanplastp.htm		
Page title	waste management and recycling		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	machine to do this	wmanirspec.htm	waste management and recycling
Non Author's	waste management and recycling		
	types of plastics		
	recyclable material		
	thermoplastics		

thermosets
polyethelene
polypropylene
ploystyrene
pvc
phenol formaldehyde
urea fomaldehyde

Page number page086
File name wmanreduc.htm
Page title waste management and recycling

	Keyword	Destination File name	Destination Page Title
Author's	designing products to be more easily recycled and repaired	wmandesign.htm	waste management and recycling
	reducing the amount of packaging	wmanpack.htm	waste management and recycling

Non Author's waste management and recycling
types of plastics
recyclable material
reducing packaging

Page number page087
File name wmanreuse.htm
Page title waste management and recycling

	Keyword	Destination File name	Destination Page Title
Author's	different sorts of waste	wmandust.htm	waste management and recycling
	waste into compost	wmancompost.htm	waste management and recycling
	finding out what plastic something is made of	wmaplasticsep.htm	waste management and recycling
	to provide the power to make other things	wmanfuel.htm	waste management and recycling
	aluminium drinks cans	wmanalurecyc.htm	waste management and recycling

Non Author's waste management and recycling
types of plastic
waste
recycling plastics
fuel incinerators

WEBSITE 2

Page number page088
File name Acid.htm
Page title acid from clouds

	Keyword	Destination File name	Destination Page Title
Author's			

Non Author's sulphur dioxide
nitrogen oxides
pH
acidic
acid rain
pollution
emission
fossil fuel
acidification
nitrogen emission
power station

Page number	page089		
File name	acid1.htm		
Page title	acid rain 2000		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	acid rain: the facts	AcidRainFacts.htm	acid rain: the facts
	acid rain on the WWW	sites.htm	acid rain 2000
Non Author's			

Page number	page090		
File name	acidrain.htm		
Page title	hot links to web sites on acid rain		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	acid rain	AcidRainE.htm	acid rain
	acid from clouds	Acid.htm	acid from clouds
	other types of air pollution	Other.htm	other types of air pollution
	imported pollution	From.htm	imported pollution
	rivers and lakes are dying	Rivers.htm	rivers and lakes are dying
	declining forest on acid soils	Forest.htm	declining forest on acid soils
	erosion and destruction of building and monuments	Erodes.htm	our history is being destroyed
	policy and action	Policy.htm	instruments and measures
Non Author's	hot links to web sites on acid rain		

Page number	page091		
File name	AcidRainE.htm		
Page title	acid rain		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	acid rain	AcidRainE.htm	acid rain
	acid from clouds	Acid.htm	acid from clouds
	other types of air pollution	Other.htm	other types of air pollution
	imported pollution	From.htm	imported pollution
	rivers and lakes are dying	Rivers.htm	rivers and lakes are dying
	declining forest on acid soils	Forest.htm	declining forest on acid soils
	our history is being destroyed	Erodes.htm	our history is being destroyed
	instruments and measures	Policy.htm	instruments and measures
Non Author's	acid rain		

Page number	page092		
File name	Erodes.htm		
Page title	our history is being destroyed		
	Keyword	**Destination File name**	**Destination Page Title**
Author's			
Non Author's	our history is being destroyed		
	buildings		
	sulphur dioxide		
	nitrogen oxide		
	ozone		
	pollutants		
	air pollution		

Page number	page093		
File name	Eurofor.htm		
Page title	more and more european forests are declining		
	Keyword	**Destination File name**	**Destination Page Title**

Appendix A

Author's

Non Author's more and more european forests are declining
air pollution
sulphur emission
sulphur deposition
trees
pollutants

Page number page094
File name AcidRainFacts.htm
Page title acid rain: the facts

	Keyword	Destination File name	Destination Page Title
Author's	causes and forms of acid rain	AcidRainFacts.htm	acid rain: the facts
	the effect of acid rain	AcidRainFacts.htm	acid rain: the facts
	the politics of acid rain	AcidRainFacts.htm	acid rain: the facts
	the role of acid rain 2000	AcidRainFacts.htm	acid rain: the facts

Non Author's acid rain: the facts
green house effect
sulphur dioxide
nitrogen oxides
pollutants
power station
sulphur emission
ozone molecules
oxygen
water
smog
acid rain
acidic
trees
acidity
ecosystem
pH
nitrogen emission
legislation
nitrogen
fresh water
lakes
power station
acidification

Page number page095
File name Forest.htm
Page title declining forests on acid soils

	Keyword	Destination File name	Destination Page Title
Author's			

Non Author's declining forests on acid soils
trees
pollution
acidic
acidification
nutrients

Page number page096
File name From.htm
Page title imported pollution

	Keyword	Destination File name	Destination Page Title
Author's			

Non Author's pollution

power station
sulphur emission
nitrogen emission

Page number page097
File name Group.htm
Page title the signing of the protocol

	Keyword	Destination File name	Destination Page Title
Author's			
Non Author's	the signing of the protocol		
	sulphur emission		
	sulphur protocol		

Page number page098
File name Longroad.htm
Page title the long road to agreements

	Keyword	Destination File name	Destination Page Title
Author's			
Non Author's	the long road to agreements		
	smoke		
	pollution		
	acid rain		
	air pollution		
	sulphur dioxide		
	ecosystems		
	rivers		
	emission		
	sulphur protocol		
	nitrogen protocol		
	acidification		

Page number page099
File name Norfor.htm
Page title norwegian forests declining

	Keyword	Destination File name	Destination Page Title
Author's			
Non Author's	norwegian forests declining		
	trees		
	acid rain		
	sulphur emission		
	nitrogen emission		
	nitrogen		
	pollution		
	acidity		
	liming		

Page number page100
File name Other.htm
Page title other types of air pollution

	Keyword	Destination File name	Destination Page Title
Author's			
Non Author's	other types of air pollution		
	acid rain		
	air pollution		
	pH		
	ozone		
	atmosphere		

smog
toxic chemicals
pollutants
ozone layer
uv radiation
cfc
greenhouse effect
greenhouse gases
carbon dioxide
nitrogen oxide
atmosphere
emission

Page number page101
File name Past.htm
Page title acid rain from prehistoric waste

	Keyword	Destination File name	Destination Page Title
Author's			
Non Author's	acid rain from prehistoric waste		
	acid rain		
	waste		
	nitrogen		
	fossil fuels		
	carbon hidrogen		
	sulphur		
	sulphur dioxide		
	nitrogen oxides		
	organic waste		
	acidification		
	pollutants		
	greenhouse effect		

Page number page102
File name Policy.htm
Page title instruments and measures

	Keyword	Destination File name	Destination Page Title
Author's			
Non Author's	instruments and measures		
	air pollution		
	acid deposition		
	legislation		
	greenhouse effect		
	emission		
	nitrogen oxides		
	sulphur dioxide		
	waste		
	gas desulphurisation		
	catalytic converter		

Page number page103
File name Respon.htm
Page title british responsibiliy

	Keyword	Destination File name	Destination Page Title
Author's			
Non Author's	british responsibiliy		
	acidification		
	emission		
	pollution		
	power station		
	sulphur emission		
	atmosphere		

nitrogen oxides
acid rain
sulphur protocol
aluminium
ferroy-alloy
nitrogen emission
catalytic converter

Page number page104
File name Rivers.htm
Page title rivers and lakes are dying

	Keyword	Destination File name	Destination Page Title
Author's			
Non Author's	river and lakes are dying		
	acid rain		
	lakes and rivers		
	acidic		
	pH		
	aluminium		
	fresh water		
	pollution		
	acid rain		
	acidification		
	nitrogen		
	acidic		
	sulphur dioxide		

Page number page105
File name Salmon.htm
Page title salmon stocks are disappearing

	Keyword	Destination File name	Destination Page Title
Author's			
Non Author's	salmon stocks are disappearing		
	sulphur emission		
	acid rain		
	pollution		
	aluminium		
	mercury		
	acidic		
	calcium		

Page number page106
File name Scale.htm
Page title the pH scale

	Keyword	Destination File name	Destination Page Title
Author's			
Non Author's	the pH scale		
	acidity		
	pH		
	acidic water		
	hydrogen		
	acidic		
	rainwater		
	calcium		
	water		

Page number page107
File name Species.htm

Page title	few species survive in acidic water		
	Keyword	**Destination File name**	**Destination Page Title**
Author's			
Non Author's	few species survive in acidic water		
	acidic water		
	lakes and rivers		
	ecosystem		
	pH		
	acid rain		
	sulphur protocol		
	calcium		
	water		
	emission		
	liming		
Page number	page108		
File name	Sulphur.htm		
Page title	the new sulphur protocol		
	Keyword	**Destination File name**	**Destination Page Title**
Author's			
Non Author's	the new sulphur protocol		
	emission		
	sulphur protocol		
	sulphur emission		
	ecosystem		
	sulphur deposition		
Page number	page109		
File name	Treat.htm		
Page title	a worldwide threat		
	Keyword	**Destination File name**	**Destination Page Title**
Author's			
Non Author's	a worldwide threat		
	emission		
	sulphur		
	pollution		
	acid rain		
	sulphur dioxide		
	nitrogen		
	nitrogen oxides		
	acidic		
	acidification		
Page number	page110		
File name	Why.htm		
Page title	why do fish die in acidic water?		
	Keyword	**Destination File name**	**Destination Page Title**
Author's			
Non Author's	why do fish die in acidic water?		
	fresh water		
	acidic water		
	pH		
	aluminium		
	oxygen		

Appendix A

WEBSITE 3

Page number page111
File name ozonedepletion.htm
Page title ozone depletion

	Keyword	Destination File name	Destination Page Title
Author's	overview	overview.htm	overview
	ozone measurements	measurements.htm	ozone measurements
	man-made cfc's	manmade.htm	man-made substances and their effects on the ozone
	cfc	industry_ozone_depletion.htm	cfc use in industry
	other sources of cfc's	other_ozone_depletion.htm	ozone depleting chemicals
	volcanic eruptions	volcanic.htm	volcanic eruptions
	natural cycles	natural.htm	natural cycles
	why is the hole over antarctica?	why_ozone_hole.htm	why the hole is over antarctica
	possible solutions	solutions.htm	possible solutions
	monitoring the ozone layer	monitoring.htm	monitoring the ozone layer
	effects of the ozone hole	effects.htm	effects of the ozone hole
	impacts on humans	impacts.htm	impacts of ozone depletion
Non Author's	ozone depletion		

Page number page112
File name overview.htm
Page title overview

	Keyword	Destination File name	Destination Page Title
Author's	ozone layer	glossary.htm	
	return to the table of contents	ozonedepletion.htm	ozone depletion
Non Author's	overview		
	particles		
	uv radiation		
	atmosphere		
	cfc		
	refrigerants		
	coolants		
	propellants		

Page number page113
File name measurement.htm
Page title ozone measurements

	Keyword	Destination File name	Destination Page Title
Author's	ozone	glossary.htm	
	return to the table of contents	ozonedepletion.htm	ozone depletion
Non Author's	ozone measurements		
	ozone layer		
	atmosphere		
	dobson unit		

Page number page114
File name manmade.htm
Page title man made cfc's

	Keyword	Destination File name	Destination Page Title
Author's	ozone	glossary.htm	
	return to the table of contents	ozonedepletion.htm	ozone depletion
Non Author's	man made cfc's		
	ozone layer		
	atmosphere		
	cfc's		
	ozone hole		

ozone depletion
volcanoes
greenhouse effect
uv radiation
stratosphere

Page number page115
File name industry_ozone_depletion.htm
Page title cfc use in industry

	Keyword	Destination File name	Destination Page Title
Author's	return to the table of contents	ozonedepletion.htm	ozone depletion
Non Author's	cfc use in industry		
	cfc's		
	the montreal protocol		
	ozone layer		
	propellants		
	aerosol		

Page number page116
File name other_ozone_depletion.htm
Page title other sources of ozone depletion

	Keyword	Destination File name	Destination Page Title
Author's	return to the table of contents	ozonedepletion.htm	ozone depletion
Non Author's	other sources of ozone depletion		
	stratosphere		
	chlorine		
	uv radiation		
	ozone depletion		
	hydrogen chloride		
	cfc's		
	ozone layer		
	oxygen		

Page number page117
File name volcanic.htm
Page title volcanic eruptions

	Keyword	Destination File name	Destination Page Title
Author's	return to the table of contents	ozonedepletion.htm	ozone depletion
Non Author's	volcanic eruptions		
	hydrogen chloride		
	stratosphere		
	carbon dioxide		
	water		
	atmosphere		
	aerosol		
	liquid fuel		

Page number page118
File name natural.htm
Page title natural cycles

	Keyword	Destination File name	Destination Page Title
Author's	return to the table of contents	ozonedepletion.htm	ozone depletion
Non Author's	natural cycles		
	ozone depletion		
	uv		
	stratosphere		
	atmosphere		
	solar activity		
	natural fluctuation		

Page number	page119		
File name	why_ozone_hole.htm		
Page title	why is the hole over antarctica?		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	return to the table of contents	ozonedepletion.htm	ozone depletion
Non Author's	why is the hole over antarctica?		
	nitrogen oxide		
	chlorine oxide		
	nitric acid		
	ozone depletion		
	atmosphere		
	ozone depleting chlorine		

Page number	page120		
File name	solutions.htm		
Page title	possible solutions		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	return to the table of contents	ozonedepletion.htm	ozone depletion
Non Author's	possible solutions		
	stratosphere		
	troposphere		
	cfc		
	ozone depleting chlorine		
	atmosphere		

Page number	page121		
File name	monitoring.htm		
Page title	monitoring the ozone layer		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	return to the table of contents	ozonedepletion.htm	ozone depletion
Non Author's	ozone layer		
	ozone depleting chemicals		
	toms		
	northern hemisphere		
	uv radiation		
	atmosphere		

Page number	page122		
File name	effects.htm		
Page title	effects of the ozone hole		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	return to the table of contents	ozonedepletion.htm	ozone depletion
Non Author's	effects of the ozone hole		
	uv radiation		
	uvb radiation		
	ozone depletion		
	antarctica		
	phytoplankton		
	atmosphere		

Pege number	page123		
File name	impacts.htm		
Page title	impacts on human		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	return to the table of contents	ozonedepletion.htm	ozone depletion
Non Author's	impacts on human		

ozone depletion
uvb radiation
uv
skin cancer

WEBSITE 4

Page number page124
File name cleanairaction.htm
Page title clean air action

	Keyword	Destination File name	Destination Page Title
Author's	facts (icon)	facts.htm	air quality facts
	tips (icon)	tips.htm	ozone tips
Non Author's	clean air action		
	health effects		
	clean air ozone level		
	ozone watch		
	ozone warning		

Page number page125
File name facts.htm
Page title air quality facts

	Keyword	Destination File name	Destination Page Title
Author's	the ozone program	oprog.htm	the ozone program
	air quality in the houston area	airq.htm	air quality in the houston area
	health effects of air pollution	healthe.htm	health effects of air pollution
	industry's contribution to regional air quality improvements	industry.htm	industry's contribution to regional air quality improvements
	why vehicle inspections are important to air quality?	vinspec.htm	why vehicle inspections are important to air quality?
	proper vehicle maintenance helps improve air quality	vehmain.htm	proper vehicle maintenance helps improve air quality
	what is planned to .....in the houston galveston region?	planned.htm	what is planned to .....in the houston galveston region?
	clean air action home page	cleanairaction.htm	clean air action
	facts (icon)	facts.htm	air quality facts
	tips (icon)	tips.htm	ozone tips
Non Author's			

Page number page126
File name oprog.htm
Page title the ozone program

	Keyword	Destination File name	Destination Page Title
Author's	facts (icon)	facts.htm	air quality facts
	tips (icon)	tips.htm	ozone tips
	clean air action home page	cleanairaction.htm	clean air action
Non Author's	the ozone program		
	ozone		
	pollutants		
	people		
	ozone level		
	air quality		
	health effects		
	ozone watch		
	ozone warning		
	ozone formation		

Page number page127
File name airq.htm
Page title air quality in the houston area

	Keyword	Destination File name	Destination Page Title
Author's	facts (icon)	facts.htm	air quality facts
	tips (icon)	tips.htm	ozone tips
	clean air action home page	cleanairaction.htm	clean air action

Non Author's
air quality in the houston area
pollutants
carbon dioxide
sulphur dioxide
nitrogen dioxide
lead
particulates
ozone level
ozone
voc
emission
smoke
air quality

Page number page128
File name healthe.htm
Page title health effects of air pollution

	Keyword	Destination File name	Destination Page Title
Author's	facts (icon)	facts.htm	air quality facts
	tips (icon)	tips.htm	ozone tips
	clean air action home page	cleanairaction.htm	clean air action

Non Author's
health effects of air pollution
air pollution
health effects
pollutants
people
health risk
air quality
ozone
particles
lead
carbon monxide
sulphur dioxide
nitrogen dioxide

Page number page129
File name industry.htm
Page title industry's contribution to regional air quality improvements

	Keyword	Destination File name	Destination Page Title
Author's	facts (icon)	facts.htm	air quality facts
	tips (icon)	tips.htm	ozone tips
	clean air action home page	cleanairaction.htm	clean air action

Non Author's
industry's contribution to regional air quality improvements
pollutants
ozone
particles
lead
carbon monoxide
sulphur dioxide
nitrogen dioxide
clean air act
voc emission
waterwaste
air quality

Page number	page130		
File name	vinspec.htm		
Page title	why vehicle inspections are important to air quality?		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	facts (icon)	facts.htm	air quality facts
	tips (icon)	tips.htm	ozone tips
	clean air action home page	cleanairaction.htm	clean air action
Non Author's	why vehicle inspections are important to air quality?		
	air pollution		
	fuel		
	vehicles		
	emission		
	hydrocarbons		
	lead		
	gas		
	clean air		

Page number	page131		
File name	vehmain.htm		
Page title	proper vehicle maintenance helps improve air quality		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	facts (icon)	facts.htm	air quality facts
	tips (icon)	tips.htm	ozone tips
	clean air action home page	cleanairaction.htm	clean air action
Non Author's	proper vehicle maintenance helps improve air quality		
	emission		
	exhaust		
	ozone		
	air pollution		
	pollutants		
	fuel		
	catalytic converter		
	oil		
	smoke		

Page number	page132		
File name	planned.htm		
Page title	what is planned to .....in the houston galveston region?		
	Keyword	**Destination File name**	**Destination Page Title**
Author's	facts (icon)	facts.htm	air quality facts
	tips (icon)	tips.htm	ozone tips
	clean air action home page	cleanairaction.htm	clean air action
Non Author's	what is planned to .....in the houston galveston region?		
	air pollution		
	catalytic converter		
	emission		
	air quality		
	ozone pollution		
	fuel vehicle		

Page number	page133
File name	tips.htm
Page title	ozone tips

Appendix A

	Keyword	Destination File name	Destination Page Title
Author's	facts (icon)	facts.htm	air quality facts
	tips (icon)	tips.htm	ozone tips
	clean air action home page	cleanairaction.htm	clean air action
Non Author's	ozone tips		
	ozone		
	clean up		
	gas		
	spill		
	air quality		
	clean air action		

A.2 Keyword Catalogue

Following the principles outlined in chapters 6 and 7, the *keyword catalogue* for the MLTutor content was constructed as follows.

```
001acid#rain#################################################################
002acid#rain:the#facts#######################################################
003acidic####################################################################
004acidic#water##############################################################
005acidification#############################################################
006acidity###################################################################
007aerosol###################################################################
008agricultural#chemicals####################################################
009air#pollution##############################################################
010air#quality################################################################
011air#quality#facts##########################################################
012air#quality#in#the#houston#area############################################
013aluminium##################################################################
014atmosphere#################################################################
015biodegradable##############################################################
016buildings##################################################################
017calcium#####################################################################
018carbon#dioxide##############################################################
019carbon#monoxide#############################################################
020catalytic#converter########################################################
021cfc#########################################################################
022cfc#use#in#industry#########################################################
023cfcs#in#the#stratosphere###################################################
024chemical#dispersants########################################################
025chemical#dispersion#########################################################
026chlorine####################################################################
027chlorofluorocarbons#########################################################
028clean#air#action############################################################
029clean#up#####################################################################
030collection##################################################################
031ddt#########################################################################
032declining#forest#on#acid#soils#############################################
033detergent#formulation######################################################
034dustbin#####################################################################
035ecosystems##################################################################
036effects#of#the#ozone#hole###################################################
037electromagnetic#fields######################################################
038emission#####################################################################
039exhaust#gases###############################################################
040exxon#valdez################################################################
041fertilisers#################################################################
042fossil#fuels################################################################
043fresh#water#################################################################
044fuel#########################################################################
045global#problem##############################################################
046greenhouse#gases############################################################
047health#effects##############################################################
048how#do#we#clean#up#oil#slicks?##############################################
049how#much#trouble#is#acid#rain?##############################################
050hydrocarbons################################################################
051hydrogen#chloride###########################################################
052imported#pollution#########################################################
053incinerator#################################################################
054infra#red#spectroscopy#####################################################
055infra-red#radiation########################################################
056instruments#and#measures###################################################
057lakes#and#rivers############################################################
058legislation#################################################################
```

059light#pollution
060liming
061methane
062natural#cycles
063natural#dispersion
064nitrogen
065nitrogen#dioxide
066nitrogen#emission
067nitrogen#oxides
068now#it's#your#turn!
069nutrients
070oil#slicks
071oil#spills
072other#sources#of#oil#pollution
073other#types#of#air#pollution
074our#history#is#being#destroyed
075overview
076oxygen
077ozone
078ozone#depleting#chemicals
079ozone#depletion
080ozone#layer
081ozone#level
082ozone#measurements
083ozone#tips
084ozone#warning
085ozone#watch
086paper#recycling
087particles
088particulates
089people
090pH
091plastic#recycling
092pollutants
093pollution
094polyethylene
095possible#solutions
096power#station
097prevention#of#oil#slicks
098problems#of#the#greenhouse#effect
099propellants
100proper#vehicle#maintenance#helps#improve#air#quality
101protocols
102radiation
103recalcitrant#molecules
104recyclable#material
105refrigerants
106smog
107smoke
108so#what's#the#problem?
109solvents
110sources#of#oil#pollution
111stratosphere
112sulphur
113sulphur#deposition
114sulphur#dioxide
115sulphur#emission
116the#aral#sea
117the#braer#oil#spill
118the#greenhouse#effect
119the#greenhouse#effect#legislation
120the#greenhouse#effect#possible#solutions
121the#gulf#war
122the#layer#of#the#atmosphere
123the#north#sea

```
124the#ozone#program#########################################################
125the#valdez#principles#######################################################
126thermal#pollution###########################################################
127toxic#metals################################################################
128trees#######################################################################
129troposphere#################################################################
130voc##########################################################################
131volcanic#eruptions##########################################################
132waste#######################################################################
133waste#management#and#recycling##############################################
134water#######################################################################
135water#pollution#############################################################
136what#can#we#do?#############################################################
137why#is#the#hole#over#antarctica?############################################
138organic#waste###############################################################
```

A.3 Attribute Database in MLTutor

The following *attribute database* was built for the MLTutor prototype using the keyword of A.2 and the page information from A.1.

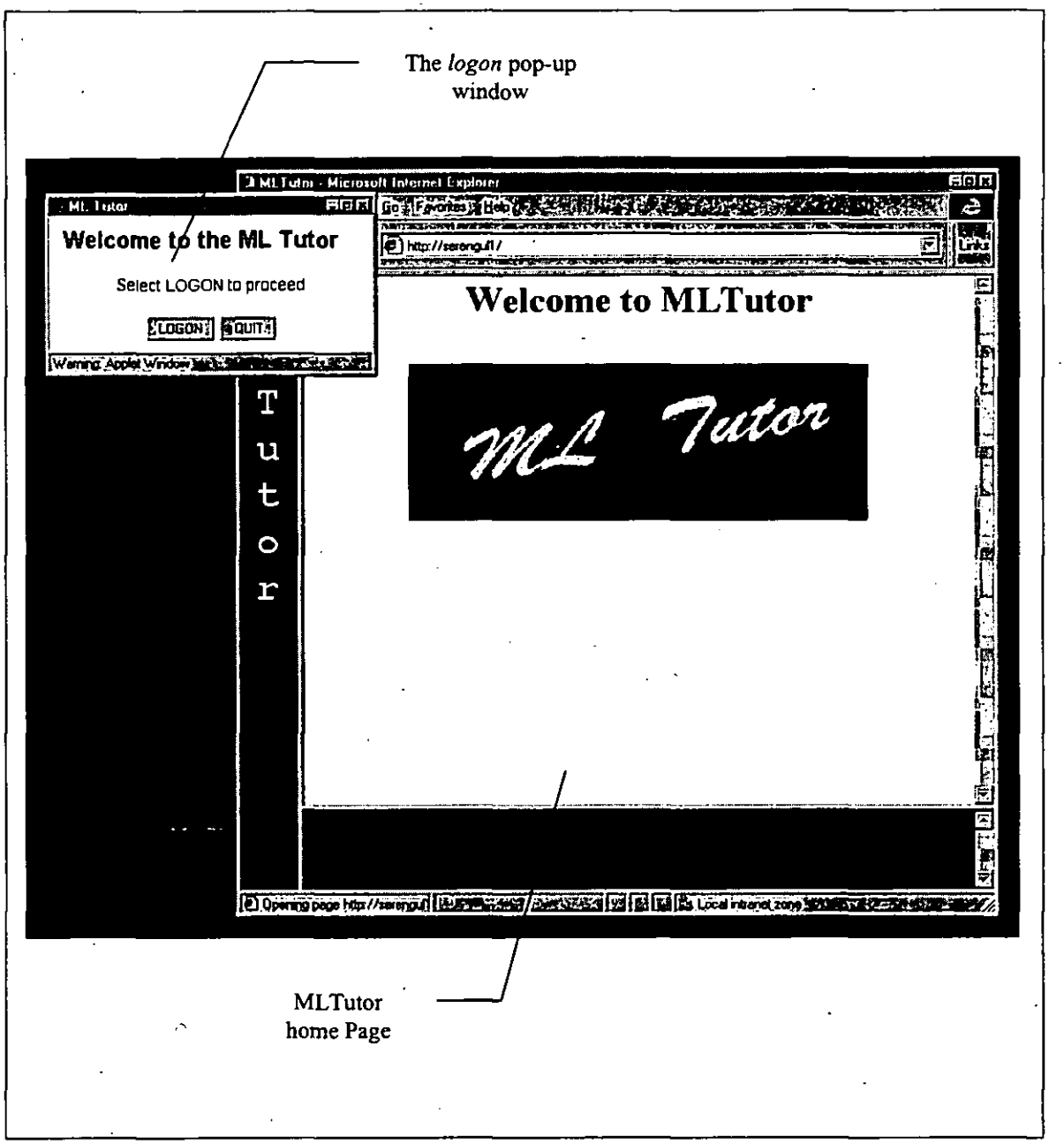
```
page000*************************************************************************************************************************************************************************
page0011000000010000001000000000000000000000000000000000100000001000000000000000000000000000000000100000000000000000000000000000000000000000001000000000000acid.buildings
page0021000000010000000100000001000000000000000000000000000000000000001001000000000000000000000000000000000000000000100000100000000000000000000000000acid.formation
page0031000000010000000000000000000000000000000000000000100000000100000000000000000000000000000000000000000000000000000000100000000000000000000000010000acid.home
page0041000000010000000000000000000000000000000000000010001000000000000000000000000000000000000000000000001000000000000000000010000000000000000000000010000acid.how.big.problem
page0050011000010001001000000000000000000000000000000000100000001000000000000000000000000000000000000110000000000000000000000000000000000011000000000000acid.lakes
page0061000000010000000000000000000000000000000000001000000000000100000000100000000000000000000000000000000000000000000000001000000000000000000000000000acid.laws
page0071010000010000001000000000000000000000000000000010100000001000000000000000000000000000000001000000000000000000000000000000000000011000000000000acid.people
page0081000000010000001100000000000000000000000000000000100000001000000000000100000000000000000000001000000000000000000000000000000000000001000000000000acid.trees
page0091010000010000000010000000100000000000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000000000000000000acid.what.is.it
page0101000000001000010001010000000000000000000000001000000000000000100000000100000000000000000000000000000001000000000000000001100100000000000000000010000acid.what.now
page0111000000010000001000000000000000000000000000000000100000001000000000000000000000000000000000001000000000000000000000001000000000000000000000000000acid.what.problem
page0121000000010000000000000000000000000000000000001000000000000000000000001000000000000000000000000000000000000000000000001000001000000000000000000001000000acid.where.from
page0131010010000000000000000000000000000000000000000000000000001000000000000000000100000000000000000000000001000000000000000001000010000000000000000000000000acid_rain
page0140000000100000000000000000000000000000100000100000000000000000000000000000000000000000000000000000000000000010000000000000000000000000000001100000agrochem
page0150000000000000000000000000000000000100000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000agrochem.ddt
page0160000000000000000100000000000000000001000000000000000000000000000000000000000000000000000000000000010000000000000000000000000000000000000000000000agrochem.detergents
page0170000000100000000000000000000000000101000000000000000000001000000000000000000000000000000000000010000000001000000000000000000000000000000001100000agrochem.recalcitrant
page0181000000010000000000000000000000000000000000000000000000000000000000000000000000100000000010000000000000000100000000000010000000000000000000000000air.pollution
page0190000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000000000000000000000000100000000000000000000001000000aral.sea
page0200000000000000000000000000100001000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000000000000case1_1
page0210000000000000000000000000000000000000000000000000000000000000000000000000001001000000000000000000000000000000000000000000000000000000000000000000case1_5
page0220000000000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000000000000case2_1
page0230000000000000000000000000100000000000000000000000000000000000000000000000001011000000000000000000000000000000000000000000000000000000000000000000case2_10
page0240000000000000000000000000100001000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000000000000case2_11
page0250000000000000000000000000100000000000000000000000000000000000000000000000001011000000000000000000000000000000000000000000000000000000000000000000case2_2
page0260000000000000000000000000100001000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000000000000case2_3
page0270000000000000000000000000000000000000000000000000000000000000000000000000001011000000000000000000000000000000000000000000000000000000000000000000case2_4
page0280000000000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000000000000case2_6
page0290000000000000000000000000000100000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000000000000case2_7
page0300000000000000000000000000000000000000000000000000000000000000000000000000001011000000000000000000000000000000000000000000000000000000000000000000case2_8
page0311000000010000000000000000010000000000000000000001000000000000000000000000000000000000000000000000000000000000110000000000000000000000000000000000domestic.smoke
page0321000000010000000010010000010000000000000001000100000000100000100000100000000010000000000000000000110000010001000000001010000000000000110000000greenhouse.causes
page0330000000010000100000000000000000000000000000000100000000100000000000000000000000000000000000000000100000000000000000001110000000000000000000000greenhouse.effect
page0340000000000000000000001000000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000001100000000000000000000000greenhouse.laws
page0350000000000000000000000000000000000100000000000000000000000000000000000000000000000000000000000100000000000000000000101000000000000000000000000greenhouse.problems
page0361000000000000000001001000000000000000000001000000000000000000000000000000000000000000000000000000000000000000000000001110000000000000000000000greenhouse.solutions
page0370000000000000000000000000000000000000011000000000000000000000100000000000000000000000000000000000001000000000000000000000000000000000000000000000LEMS
page0380000000000000000000000000000000000000000100000000000000000000001000000000000000000000000000000000100000000000000000000000000000000000000000000000Light
page0390000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000100000000000000000000000000000000000000000000000LSol
page0400000000001000000000000000000000000010000000001000000000010000000000000000000000000000000000000001000000000100000000000000000000000000000000000000LSun
page0411000000010000000000000000000000000000000000000000000000000011000000000000000000010000000000000000000000000000000000100100000000100000north.sea
page0420000000000000000000000000100001000000000000000000000000000000011000000000000000000000000000000000000000000000100000000000000000000000000oil.braer
page0430000000000000000000000000000000000000000000000000000000000011000000000000000000010000000000000000000000000000000000000000000001000000oil.chemical.dispersant
page0440000000000000000000000000100001000000000000000000000000000001000001100000000000000000000000000000000000000000000000000000000000000000000oil.cleanup
page0450000000000000000000000000100000100000000000000000000000000001000000110000000000000000000000000000000000000001000000000001000000000000001000000oil.collect
page0460000000000000000000000000000000000000000001000000000000000000000000000001000000000000000000000000000000000000000000000100000000000000000oil.exxon
page0471000000000000000000000000000000000000001000000000000000000000000000000001000000000001000000000000001000000000001000000100000001110000000oil.gulf.war
page0481000000000000000000000000000000000000001000000100000000000000000000000110000000000000000000000000000000000000001000010001000000000000001000000oil.gulf.war
page0490000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000100000000000000000000000oil.kuwait
page0500000000000000000000000000000000000000001000000000000000000000000000000001000000000000000000000000000100000000000000000000000000000000000000000000oil.measures
```

page0700000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000011000000rubbish
page0710000000001000000000010000000000000000000000000000001000000000000000100000000000000000000001000000000000000010000000000000000000000000000000smog
page0720000000000000000000000000000000000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000100000001100000thermal
page0730000000010000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000001000001000000100100000101100000water
page0740000000000000100000000000000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000010000000wmanalurecyc
page0750000000000000000000000000000000000000001000000000000000000000000000000100000000000000000000000000000000000000000000000000011000000wmancompost
page0760000000000000000000000000000000000000000000000000000000000000000000000000000000000000000100000000000000000000000000000010000000wmancompost
page0770000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000010000000wmandust
page0780000000000000000000000000000001000000000000000000000000000000000000000000000000000000001000000000000000000000000000000011000000wmanfuel
page0791000000001000000000000000000000000100001000000000000001000000000000000000000000000000000000000000000000000000000000000000011000000wmanincin
page0801000000001000000000000000000000000100001000000000000001000000000000000000000000000000000000000000000000000000000000000000011000000wmanintro
page0810000000000000000000000000000000000000000000000001000000000000000000000000000000000000100000000000000000000000000000000000000010000000wmanirspec
page0820000000000000001000000000000000000000000000000001000000000000000000000000000000001000000000000100000000000000000000000000011000000wmanpack
page0830000000000000000100000000000000000000000000001000000000000000000000000000000000000001000000000000000010000000000000100000000010000100000wmanpaperecyc
page0840000000000000000000000000000000000000000000000000000000000000000000000000000000100000000000010000000000000000000000000000000010000000wmanplasticsep
page0850000000000000000000000000000000000000000000000000000000000000000000000000000000100100000000010000000000000000000000000000000010000000wmanplastp
page0860000000000000000000000000000000000000000000000000000000000000000000000000000000100000000000010000000000000000000000000000000010000000wmanreduc
page0870000000000000000000000000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000000011000000wmanreuse
page0881010100000000000000000000000000000001000100000000000000000000000011000000000000000000000100100100000000000000000100000000000000000000000000Acid
page0891100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000acid1
page0901000000000000000000000000000000001000000000000000000001000110000000000000000110000000000000000000000000000000000000000000000000000000acidrain
page0911000000000000000000000000000000001000000000000000000001000110000000000000000110000000000000000000000000000000000000000000000000000000AcidRainE
page0920000000001000000100000000000000000000000000000000000000000100000010010000000000000010000000000000000000000100000000000000000000000000Erodes
page0930000000001000000000000000000000100000000000000000000000000000000000000000000000100000000000000000000101000000000000100000000000000Eurofor
page0941110110000000000000000000000000010000000100000000000001100000101100000000100000000000010100010000000000100000001100100000000001000001000000AcidRainFacts
page0950010100000000000000000000000000100000000000000000000000000000000000010000000000000000000000100000000000000000000000000000000001000000000000Forest
page0960000000000000000000000000000000000000000000000000000000001000000000000000000000000001001000000000000000000100000000000000000000000000From
page0970000000000000000000000000000000000000000000000000000000000000000000000000000000000000000100000000000001000000000000000000000000000Group
page0981000100000000000000000000000000100100000000000000000000001000000000000000000000000000010000001000000000000100000000000000000000000000Longroad
page0991000010000000000000000000000001000000000000000000000000001000101000000000000000000000000010000000000000000000000000000000000000000000Norfor
page1001000000001000010001001000000000000000010000000100000000000000000001000000000000000000000001000000000010001000000000000100000000100000000000000Other
page1011000100000000000000000000000000000000001000000000000000000001001000000000000000000000000000000000000000000000101000000000000000000100000000Past
page1020000000001000000000010000000000000000010000000000000000001010000000010000000000000000000000000000000000001000100000000000000100000000Policy
page1031000100000001100000100000000000000000100000000000000000000000000011000000000000000000000000010010000100000000000010000000000000000000000000Respon
page1041010100000001000000000000000000000000010000000000000100000010000000000000000000000000100100000000000000000000100000000000000000000000000000Rivers
page1051011000000001001000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000100000000000000000000000000000Salmon
page1060011010000000000010000000000000000000000000000000000000000000000000000000000000000000010000000000000000000000000000000000000000001000000Scale
page1071001000000000000100000000000000000100100000000000000000001001000000000000000000000000000000000100000000001000000000000000000000000001000000Species
page1080000000000000000000000000000000000100100000000000000000000000000000000000000000000000000000000001000000000001010000000000000000000000000000Sulphur
page1091010100000000000000000000000000000010000000000000000000000001001000000000000000000000000000100000000000000000101000000000000000000000000000Treat
page1100000100000000100000000000000000000000000000000010000000000000000000000000000000010000000000001000000000000000000000000000000000000000000000Why
page1110000000000000000000001000000000000001000000000000000000000000000001000000000000100110010000000000000100000000000000000000000000000100000100000ozonedepletion
page1120000000000000010000001000000000000000000000000000000000000000000001000100000001000000000000100100100000000000000000000000000000000000000overview
page1130000000000000010000000000000000000000000000000000000000000000000000110100000000000000000000000000000000000000000000000000000000000000000000measurement
page1140000000000000010000001000000000000000000000000000000000000000000001100000000000000000000000100000000100000010000000000001000001000000manmade
page1150000001000000000000001100000000000000000000000000000000000000000000001100000000000000000001010000000000000000000000000000000000000000000000000000industry_ozone_depletion
page1160000000000000000000000100001000000000000000000000000010000000000000000000000010011000000000000000000000010000000001000000000000000000000000000000other_ozone_depletion

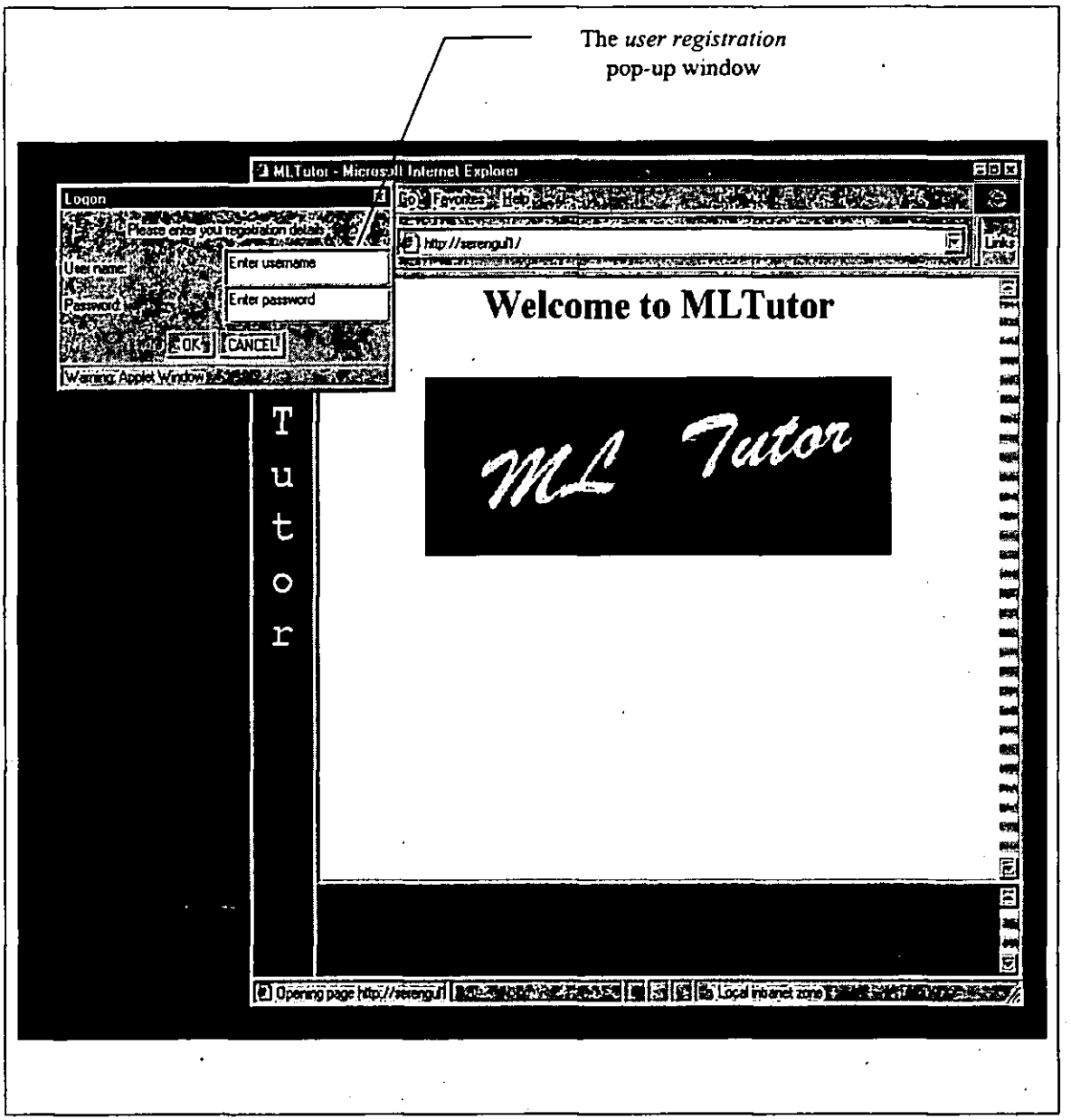
Appendix B

Appendix B

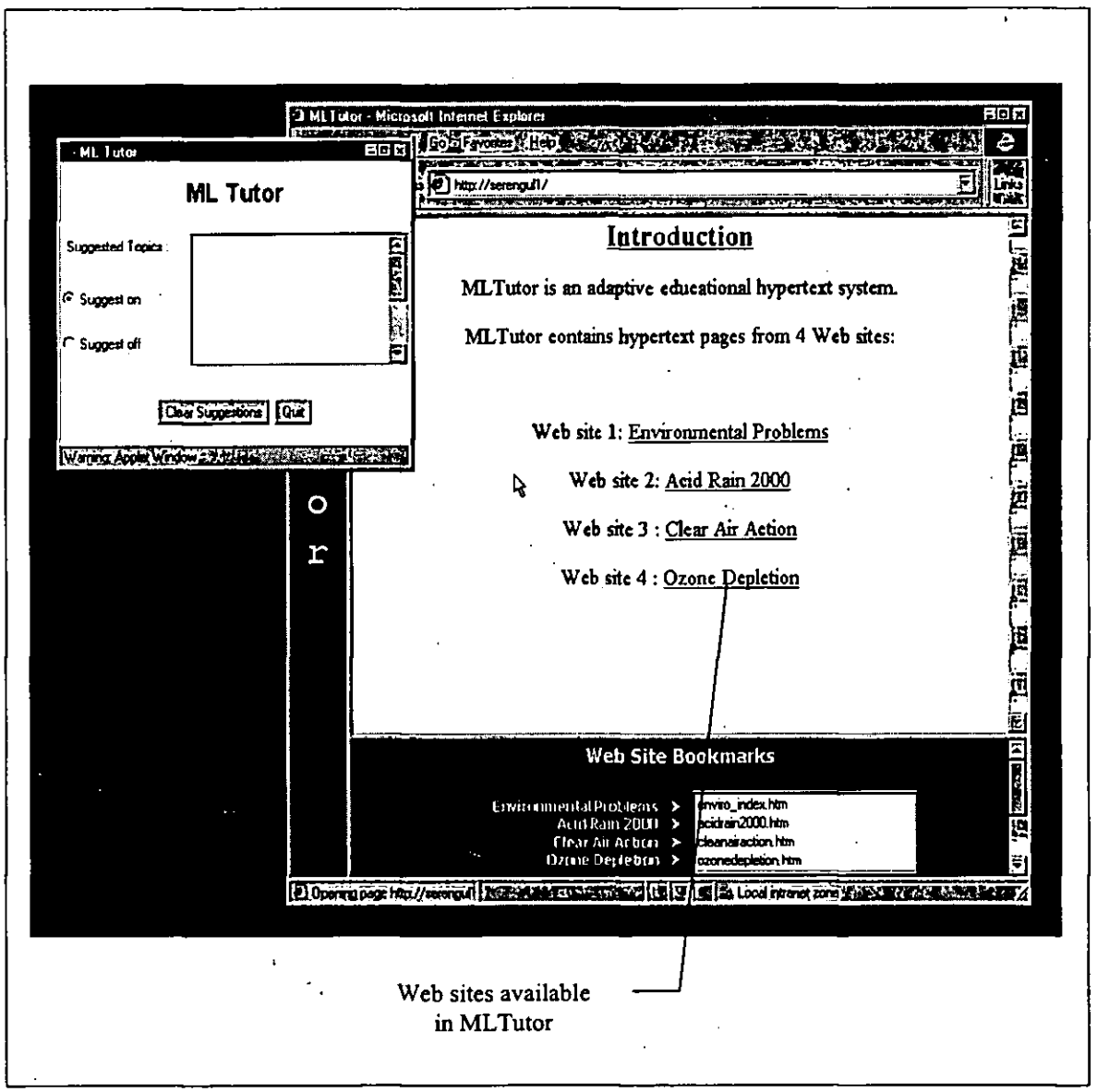
This appendix contains a sequence of MLTutor screen shots cover the functional components of the MLTutor interface.



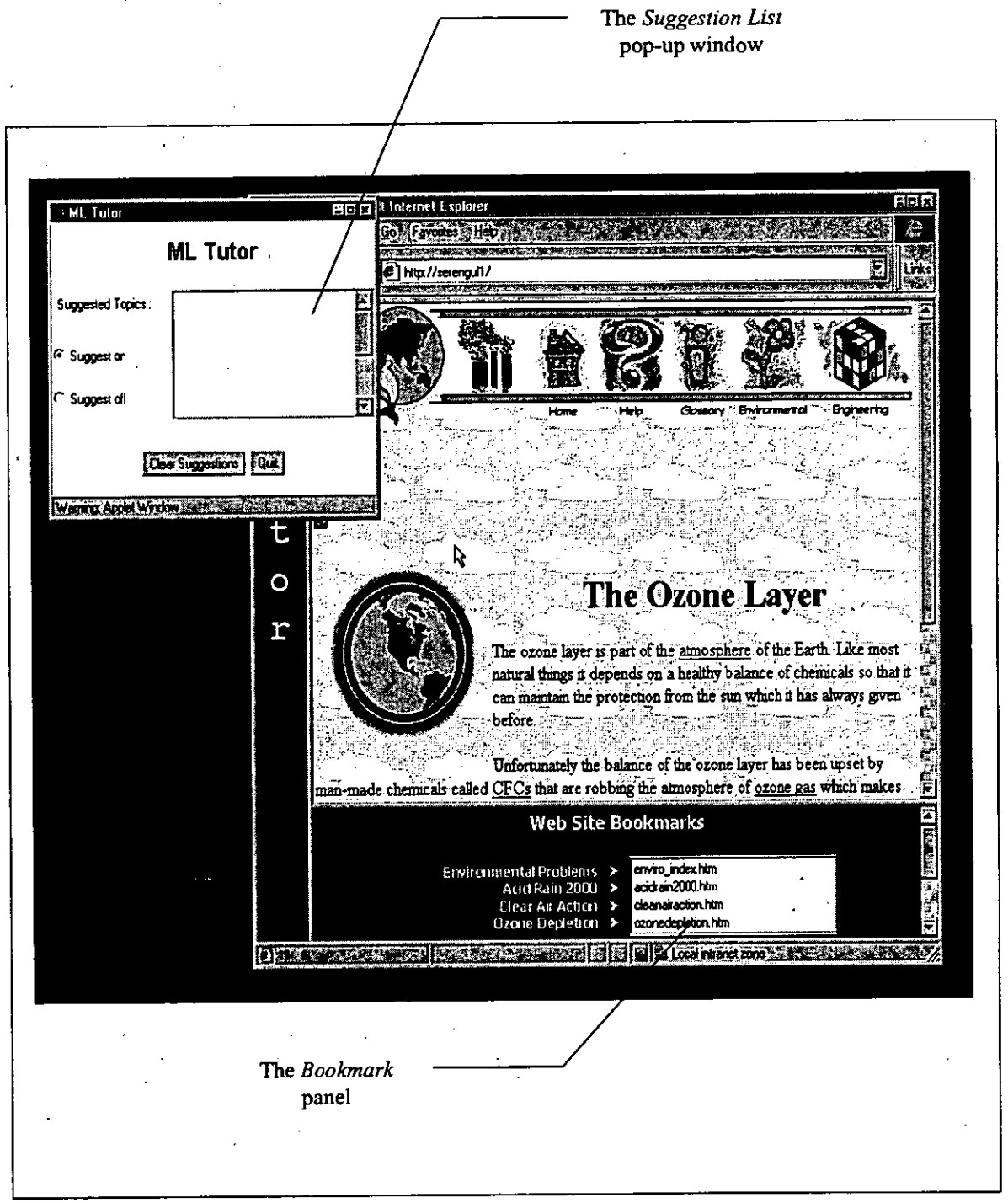
B.1: The MLTutor home page featuring the *logon* pop-up window.



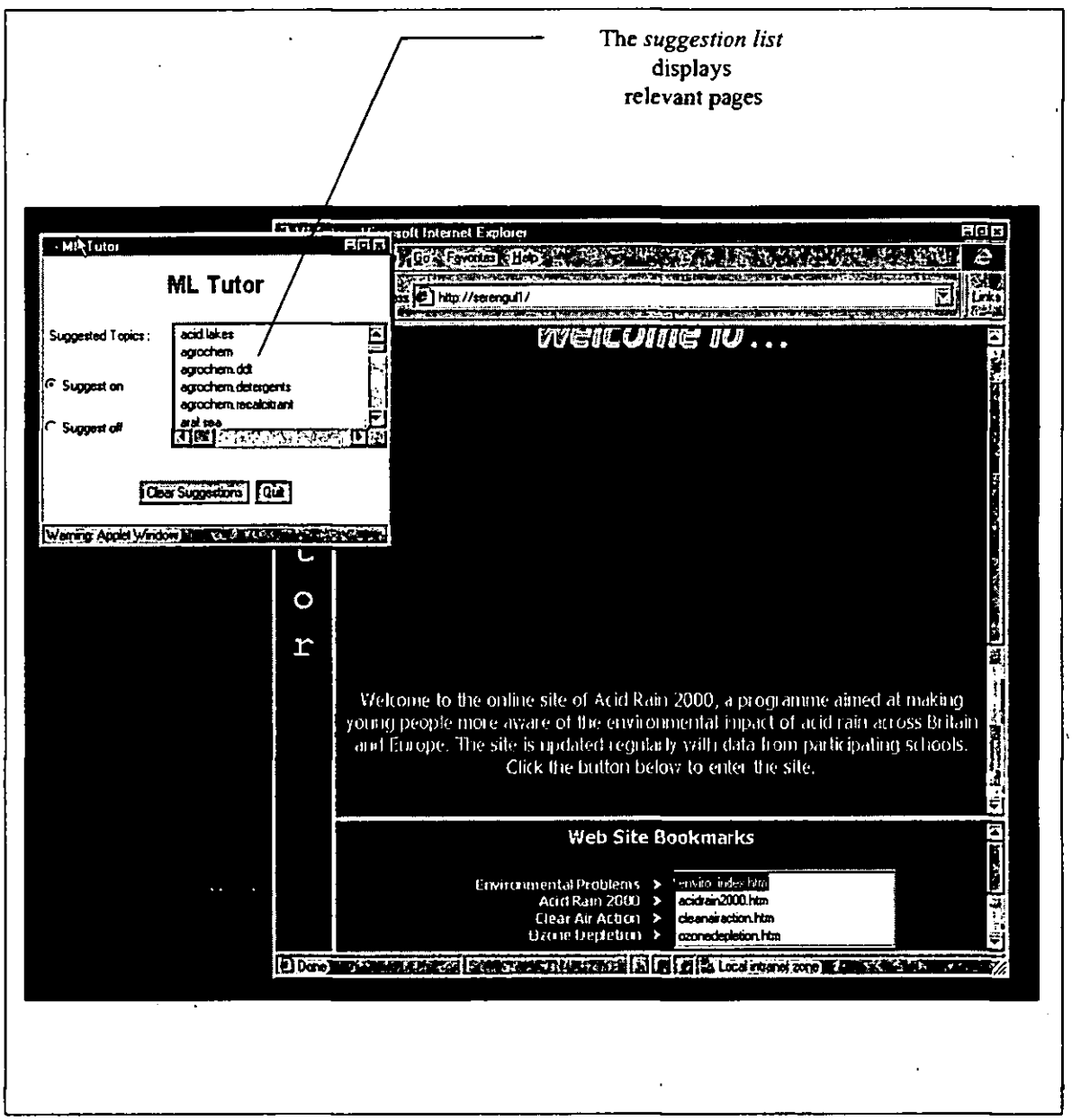
B.2: The MLTutor home page featuring the *user registration* pop-up window. Selecting the *logon* button seen in B.1 reveals this pop-up window.



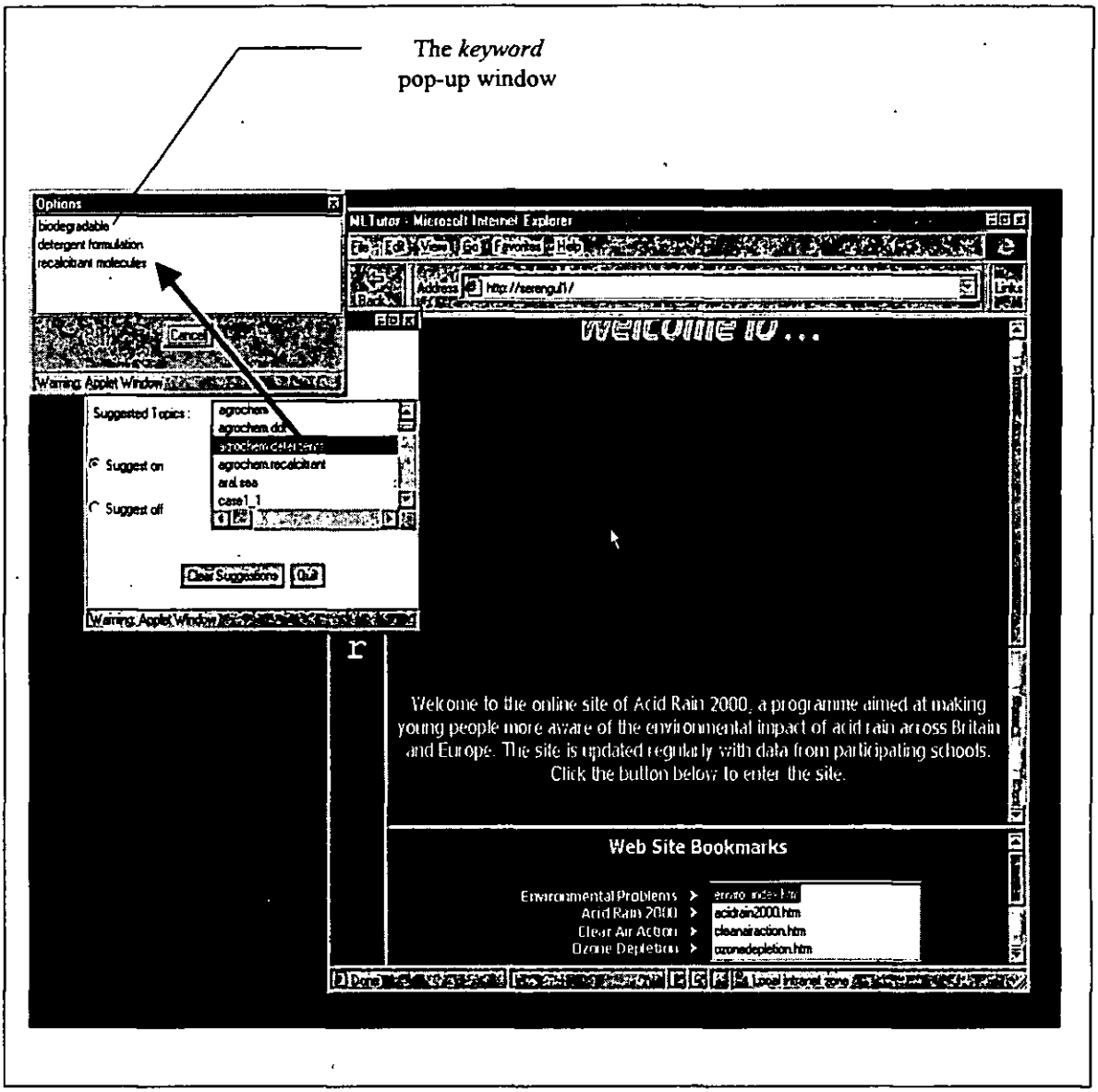
B.3: The MLTutor site selection screen featuring the suggestion list pop-up and the bookmark panel are displayed following successful user username and password entry. The suggestion list is initially blank.



B.4: The main MLTutor window displays the WWW documents. The bookmark panel maintains the list of sites available in the system.



B.5: The suggestion list pop-up window contains a list of relevant pages.



B.6: Selection of an entry from the suggestion list reveals the keyword pop-up window. This lists the keyword contained on the selected page. Selecting any keyword results in the page from the suggestion list being displayed in the main MLTutor window. The bookmark panel allows direct access to the Websites. Each Web site is entered via the site home page.

Appendix C

Appendix C

The information contained in this appendix relates to the empirical study conducted to assess the MLTutor system. Full details of the evaluation strategies can be found in chapter 8 and 9.

C.1 User Instructions

All participants who took part in the empirical study were given the following instructions prior to staring the experiment. A variant of the following user instructions, from which some sections describing the adaptive features were excluded, was also provided for the control users.

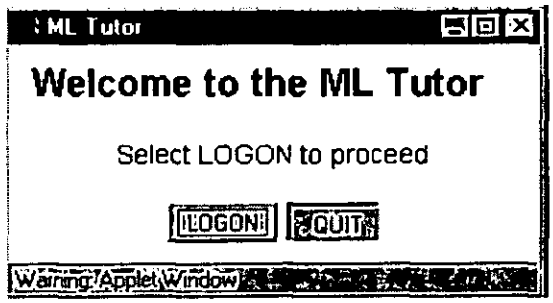
MLTutor

Thank you for agreeing to participate in this experiment.

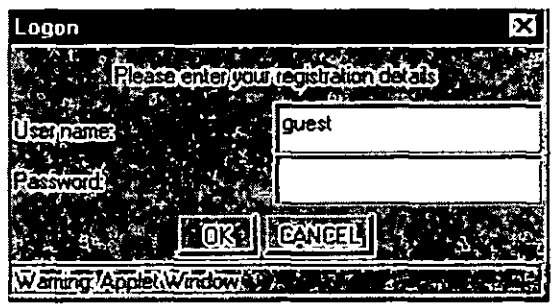
MLTutor is a hypertext-based educational system that aims to provide personal guidance to its users. Your contributions will be invaluable in our evaluation of the system. The effectiveness of the system will be measured in terms of your assessment of the assistance it provides you.

During this exercise you will be asked to complete a set of tasks using MLTutor. The educational material contained within MLTutor consists of 133 hypertext pages from four different sites on the WWW. The subject matter of these sites is 'Environmental Science'.

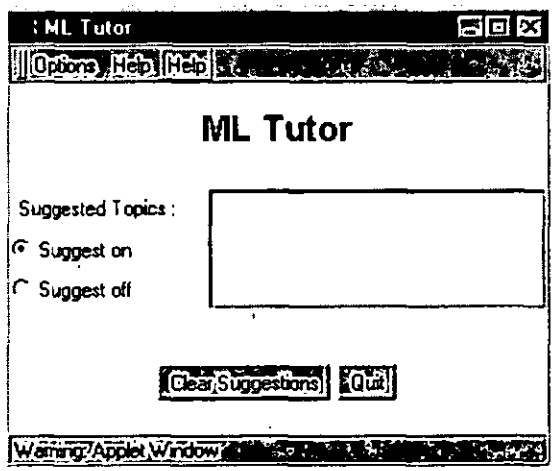
User Instructions



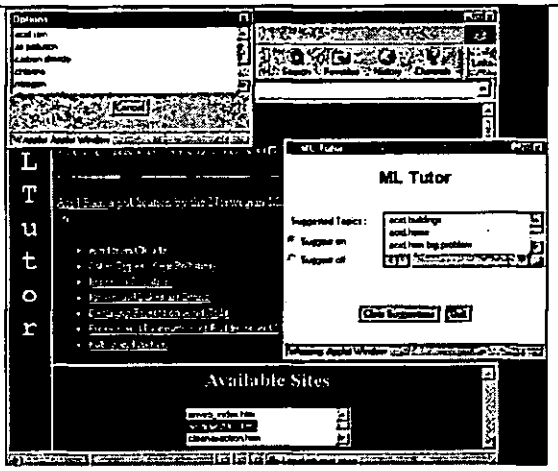
Please click the **LOGON** button to continue.



Please type '**guest**' as your username and blank out the password then click the 'OK' button to continue.



Initially the suggestion list will be blank.

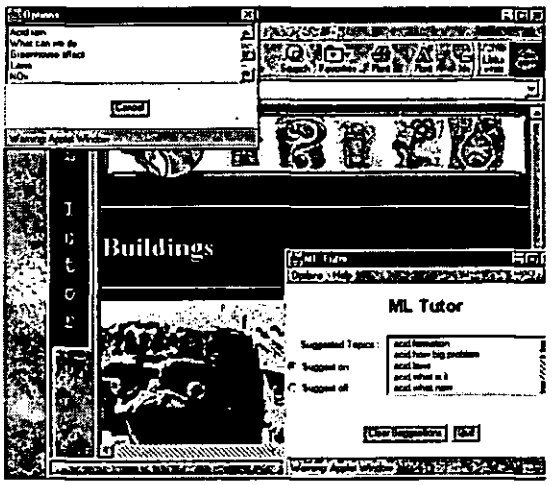


MLTutor system provides three types of navigational aids:

Web site Bookmarks – located at the bottom of every web page and allow direct access to the selected web sites.

Hotlinks – underlined words and icons on each web page allow a jump to other pages with the site.

Suggestion List – displays recommended list of pages across all sites.



The suggestion list will be updated by the system periodically.

Selecting an item in the suggestion list reveals the content related keywords in the pop-up window at the top left corner. Clicking on any keyword will take you to the suggested page.

C.2 Users' Tasks

An expert in the field of environmental science prepared the following set of tasks based on the teaching material covered in MLTutor system.

QUESTIONS

1. An aluminium company (ALGREAT) and an environmental pressure group (GREENPOWER) are debating the effects of metals on quality of life.

   ALGREAT say:
   "Aluminium is a valuable and versatile metal which is very abundant in the earth's crust and therefore causes little environmental damage."

   GREENPOWER say:
   "Aluminium is an expensive toxic metal which is damaging to the environment."

   Show how both sentences are correct by answering the following questions

   a. Why do we recycle aluminium?

   .....
   .....
   .....
   .....

   b. How do we recycle aluminium?

   .....
   .....
   .....
   .....

   c. Explain how aluminium combines with acid rain to suffocate fish and kill trees.

   .....
   .....
   .....
   .....

2. Name all the atmospheric chemicals involved in the formation of nitric and sulphuric acids (acid rain).

   .....
   .....
   .....
   .....
   .....
   .....

3. What are the differences between ground level ozone and stratospheric ozone (the "ozone layer") ?

4. How is ground level ozone produced?

5. What are the health and environmental effects of ground level ozone?

6. How can we minimise the formation of ground level ozone?

7 What systems are used by government to protect the public from the potential adverse health effects of ground level ozone?

8. What is ozone layer depletion?

9. What evidence is there for ozone layer depletion?

10. Identify the pollutants that cause ozone layer pollution.

11. What are the main sources of ozone depleting chemicals?

12. Where is the main ozone hole located and why is it found there?

13. What are the main health and environmental effects of stratospheric ozone depletion?

C.3 User Feedback Questionnaire

In order to capture the participants' perception of the MLTutor system, the following questionnaire was given to each user to complete.

Personal details

Name ..... ***mail address*** .....

Course ..... **Year of study** .....

Users' comments on MLTutor

Instructions: For each question below mark an 'X' on the line between the two extremes to indicate your answer.

poor ——|——|——|——|——|——|——|——X——|—— good

1. How easy was the MLTutor system to use?

difficult ——|——|——|——|——|——|——|——|——|—— easy

2. Did MLTutor help you with the tasks?

hindrance ——|——|——|——|——|——|——|——|——|—— helpful

3. How frequently did you use the suggestion list?

rarely ——|——|——|——|——|——|——|——|——|—— often

4. Did you find the system suggestions relevant?

irrelevant ——|——|——|——|——|——|——|——|——|—— relevant

5. How would you rate the usefulness of the system suggestions?

poor ——|——|——|——|——|——|——|——|——|—— good

6. How frequently did you use the web site bookmarks?

rarely ——|——|——|——|——|——|——|——|——|—— often

7. What did you like about the MLTutor system?

Please give details:

.....
.....
.....

8. What didn't you like about the system?

Please give details:

.....
.....
.....

9. Could you suggestion any improvements?

Please give details:

.....
.....
.....

10 Any other comments?

Please give details:

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Thank you for your co-operation

Acknowledgement

I would like to thank to Dr Ian Williams for his invaluable assistance in the preparation of this experiment.

C. 4 The domain expert recommendations

The following table displays the expert recommendations to answer each question.

Questions	Page number	Website
1a	074	Website 1
	077	Website 1
1b	074	Website 1
	087	Website 1
1c	005	Website 1
	094	Website 2
	095	Website 2
	104	Website 2
	109	Website 2
	110	Website 2
2	002	Website 1
	012	Website 1
	088	Website 2
	101	Website 2
	Full answer on 094	Website 2
3	062	Website 1
	112	Website 3
	126	Website 4
	Main answer on 063	Website 1
4	063	Website 1
	127	Website 4
5	126	Website 4
	127	Website 4
	128	Website 4
6	126	Website 4
	127	Website 4
	133	Website 4
7	121	Website 3
	126	Website 4
	129	Website 4

	130	Website 4
	132	Website 4
8	061	Website 1
	112	Website 3
	116	Website 3
9	114	Website 3
10	061	Website 1
	114	Website 3
11	114	Website 3
	115	Website 3
	116	Website 3
12	119	Website 3
13	062	Website 1
	122	Website 3
	123	Website 3

Appendix D

Appendix D

This appendix contains several tables. The tables contain data from the empirical study and summary of statistics. Full details of the analysis conducted on the data and conclusion can be found in Chapter 9.

Participant Number	Group No	Student Name	MLTutor Version
Participant 1	Group 1	Env. Sci. 3rd year	0
Participant 2	Group 2	Env. Sci. 3rd year	0
Participant 3	Group 3	Env. Sci. PhD	0
Participant 4	Group 4	Env. Sci. 3rd year	0
Participant 5	Group 5	Com. Sci. Researcher	0
Participant 6	Group 6	Comp. Sci. PhD	0
Participant 7	Group 1	Env. Sci. 3rd year	1
Participant 8	Group 2	Env. Sci. 3rd year	1
Participant 9	Group 3	Geo. Reseracher	1
Participant 10	Group 4	Env. Sci. PhD	1
Participant 11	Group 5	Env. Sci. PhD	1
Participant 12	Group 6	Comp. Sci. PhD	1
Participant 13	Group 1	Env. Sci. 3rd year	2
Participant 14	Group 2	Env. Sci. 3rd year	2
Participant 15	Group 3	Struc. Eng.	2
Participant 16	Group 4	Env. Sci. PhD	2
Participant 17	Group 5	Comp. Sci. PhD	2
Participant 18	Group 6	Comp. Sci. PhD	2
Participant 19	Group 1	Env. Sci. 3rd year	3
Participant 20	Group 2	Env. Sci. 3rd year	3
Participant 21	Group 3	Geo. PhD	3
Participant 22	Group 4	Env. Sci. PhD	3
Participant 23	Group 5	Comp. Sci. PhD	3
Participant 24	Group 6	Comp. Sci. PhD	3
Participant 25	Group 1	Env. Sci. 3rd year	4
Participant 26	Group 2	Env. Sci. 3rd year	4
Participant 27	Group 3	Geo. PhD	4
Participant 28	Group 4	Env. Sci. 3rd year	4
Participant 29	Group 5	Com. Sci. Researcher	4
Participant 30	Group 6	Comp. Sci. PhD	4

Table D.1.1: Participant details are shown in this table.

Group No	Participant No	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	Scores
Group 1	Participant 1	5	2	2	0.5	2	2	2	1	1	0.5	2	1	1	22
Group 1	Participant 7	4	3	3	1	1.5	0.25	0	1.5	1	1.5	2	1	3	22.75
Group 1	Participant 13	5.5	2	2.5	1.5	2	0.5	3	0.5	0	1	1	1	2	22.5
Group 1	Participant 19	4.5	1	1	0	0	0	0	1	0.5	1	1	1	2	13
Group 1	Participant 25	3	2	3	1.5	1	1	3	2	1	1	2	1	2	23.5
Group 2	Participant 2	5	2	2	1	1	0.25	1	1	0.5	0.5	1	1	1	17.25
Group 2	Participant 8	5.5	2	2.5	2	1	2	0.5	1	1	1	2	1	2	23.5
Group 2	Participant 14	5.5	2	3	0.5	1	2.5	1	0.5	0.5	1	2.5	0.25	3	23.25
Group 2	Participant 20	5	2	3	1	1	2	1	2	0.5	1	1	1	2	22.5
Group 2	Participant 26	3.5	2.5	1	0	0	0	0	1	1	1	1	1	1	13
Group 3	Participant 3	3.5	2	1	1	0.5	1	1	1	0.5	0.25	0.25	0.5	0	12.5
Group 3	Participant 9	5	1	2	1	1	0.5	2	1	0.5	1	1	1	1	18
Group 3	Participant 15	6	2	2	1	0.5	1.5	0	1	0.5	0.5	2	1	2	20
Group 3	Participant 21	4	3	3	1	1	1	1	0.5	0.5	1	2	1	1	20
Group 3	Participant 27	3	1	0	0	0	0	0	1	1	1	2	1	1	11
Group 4	Participant 4	5	2	3	1	1	1	1	2	0.5	2	2	1	2	23.5
Group 4	Participant 10	5	2	2.5	1	0	0	0.5	1	0.5	1	2	1	1	17.5
Group 4	Participant 16	5	2	2	1	2	2	2	0	1	1	2	1	1	22
Group 4	Participant 22	4.5	2	3	1	2	2	0	2	1	1	2	1	2	23.5
Group 4	Participant 28	5	2	1	0	2	2	0.5	1	1	0.5	2	1	2	20
Group 5	Participant 5	4	1	2	0	1	0.5	2	1	1	1	2	1	1	17.5
Group 5	Participant 11	4	2	2	1	1	2	1	1	1	1	2.5	1	1	20.5
Group 5	Participant 17	3	2	2	1	1	1	0	1	0.5	2	3	1	1	18.5
Group 5	Participant 23	4	2	1	0.5	0.5	0.5	1	1	0.5	1	1	1	2	16
Group 5	Participant 29	4	3	0	0	1.5	0	0	1	1	1	2	1	0	14.5
Group 6	Participant 6	5	2	1.5	0	1.5	0.25	0	1	0.5	1	1	1	0.25	15
Group 6	Participant 12	5	2	3	1	2	2	3	1	1	0.5	2.5	1	2	26
Group 6	Participant 18	5	2	2.5	0	0	0	1.5	2	0	1	1	1	2	18
Group 6	Participant 24	5	2	4	2	2.5	1	3	0.5	0	0.5	2	1	2.5	26
Group 6	Participant 30	3.5	2	1	0	0	0.25	2	1	0.5	0.5	1	1	1.5	14.25

Table D.2.1: Participant scores to the tasks shown in C.2, marked as per the recommended solutions in C.4, are shown. The data categorised on participant groups.

Group No	Mean	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	Score
1	mean	4.4	2	2.3	0.9	1.3	0.8	1.6	1.2	0.7	1	1.6	1	2	20.8
	stdev	3.7	2	2.8	1.7	2.8	2.5	9.2	1.3	0.8	0.5	1.2	0	2	4.4
2	mean	4.9	2.1	2.3	0.9	0.8	1.4	0.7	1.1	0.7	0.9	1.5	0.9	1.8	19.9
	stdev	2.7	0.2	2.8	2.2	0.8	5.2	0.8	1.2	0.3	0.2	2	0.5	2.8	4.6
3	mean	4.3	1.8	1.6	0.8	0.6	0.8	0.8	0.9	0.6	0.8	1.5	0.9	1	16.3
	stdev	5.8	2.8	5.2	0.8	0.7	1.3	2.8	0.2	0.2	0.5	2.6	0.2	2	4.3
4	mean	4.9	2	2.3	0.8	1.4	1.4	0.8	1.2	0.8	1.1	2	1	1.6	21.3
	stdev	0.2	0	2.8	0.8	3.2	3.2	2.3	2.8	0.3	1.2	0	0	1.2	2.6
5	mean	3.8	2	1.4	0.5	1	0.8	0.8	1	0.8	1.2	2.1	1	1	17.4
	stdev	0.8	2	3.2	1	0.5	2.3	2.8	0	0.3	0.8	2.2	0	2	2.3
6	mean	4.7	2	2.4	0.6	1.2	0.7	1.9	1.1	0.4	0.7	1.5	1	1.7	19.9
	stdev	1.8	0	5.7	3.2	5.3	2.7	6.2	1.2	0.7	0.3	2	0	3	5.8

Table D.2.2: Containing summary statistics for the data in D.2.1.

Version	Participant Number	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	Score
0	Participant 1	5	2	2	0.5	2	2	2	1	1	0.5	2	1	1	22
	Participant 2	5	2	2	1	1	0.25	1	1	0.5	0.5	1	1	1	17.25
	Participant 3	3.5	2	1	1	0.5	1	1	1	0.5	0.25	0.25	0.5	0	12.5
	Participant 4	5	2	3	1	1	1	1	2	0.5	2	2	1	2	23.5
	Participant 5	4	1	2	0	1	0.5	2	1	1	1	2	1	1	17.5
	Participant 6	5	2	1.5	0	1.5	0.25	0	1	0.5	1	1	1	0.25	15
1	Participant 7	4	3	3	1	1.5	0.25	0	1.5	1	1.5	2	1	3	22.75
	Participant 8	5.5	2	2.5	2	1	2	0.5	1	1	1	2	1	2	23.5
	Participant 9	5	1	2	1	1	0.5	2	1	0.5	1	1	1	1	18
	Participant 10	5	2	2.5	1	0	0	0.5	1	0.5	1	2	1	1	17.5
	Participant 11	4	2	2	1	1	2	1	1	1	1	2.5	1	1	20.5
	Participant 12	5	2	3	1	2	2	3	1	1	0.5	2.5	1	2	26
2	Participant 13	5.5	2	2.5	1.5	2	0.5	3	0.5	0	1	1	1	2	22.5
	Participant 14	5.5	2	3	0.5	1	2.5	1	0.5	0.5	1	2.5	0.25	3	23.25
	Participant 15	6	2	2	1	0.5	1.5	0	1	0.5	0.5	2	1	2	20
	Participant 16	5	2	2	1	2	2	2	0	1	1	2	1	1	22
	Participant 17	3	2	2	1	1	1	0	1	0.5	2	3	1	1	18.5
	Participant 18	5	2	2.5	0	0	0	1.5	2	0	1	1	1	2	18
3	Participant 19	4.5	1	1	0	0	0	0	1	0.5	1	1	1	2	13
	Participant 20	5	2	3	1	1	2	1	2	0.5	1	1	1	2	22.5
	Participant 21	4	3	3	1	1	1	1	0.5	0.5	1	2	1	1	20
	Participant 22	4.5	2	3	1	2	2	0	2	1	1	2	1	2	23.5
	Participant 23	4	2	1	0.5	0.5	0.5	1	1	0.5	1	1	1	2	16
	Participant 24	5	2	4	2	2.5	1	3	0.5	0	0.5	2	1	2.5	26
4	Participant 25	3	2	3	1.5	1	1	3	2	1	1	2	1	2	23.5
	Participant 26	3.5	2.5	1	0	0	0	0	1	1	1	1	1	1	13
	Participant 27	3	1	0	0	0	0	0	1	1	1	2	1	1	11
	Participant 28	5	2	1	0	2	2	0.5	1	1	0.5	2	1	2	20
	Participant 29	4	3	0	0	1.5	0	0	1	1	1	2	1	0	14.5
	Participant 30	3.5	2	1	0	0	0.25	2	1	0.5	0.5	1	1	1.5	14.25

Table D.3.1: The data of table D.2.1 is re-categorised based on MLTutor versions used.

MLTutor Version	Mean	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	Score
0	mean	4.6	1.8	1.9	0.6	1.2	0.8	1.2	1.2	0.7	0.9	1.4	1	0.9	18.0
	stdev	0.7	0.4	0.7	0.5	0.5	0.7	0.8	0.4	0.3	0.6	0.7	0.2	0.7	4.2
1	mean	4.8	2	2.5	1.2	1.1	1.1	1.2	1.1	0.8	1	2	1	1.7	21.4
	stdev	0.6	0.6	0.4	0.4	0.7	1	1.1	0.2	0.3	0.3	0.5	0	0.9	3.3
2	mean	5	2	2.3	0.8	1.1	1.3	1.3	0.8	0.4	1.1	1.9	0.9	1.8	20.7
	stdev	1	0	0.4	0.5	0.8	0.9	1.2	0.7	0.4	0.5	0.8	0.3	0.8	2.2
3	mean	4.5	2	2.5	0.9	1.2	1.1	1	1.2	0.5	0.9	1.5	1	1.9	20.2
	stdev	0.4	0.6	1.2	0.7	0.9	0.8	1.1	0.7	0.3	0.2	0.5	0	0.5	4.9
4	mean	3.7	2.1	1	0.3	0.8	0.5	0.9	1.2	0.9	0.8	1.7	1	1.3	18.0
	stdev	0.8	0.7	1.1	0.6	0.9	0.8	1.3	0.4	0.2	0.3	0.5	0	0.8	4.7

Table D.3.2: Containing summary statistics for the data in D.3.1.

MLTutor Version	Participant Number	Time spent	Built-in links visited	Bookmarks visited	Suggestion list links visited	Web sites visited
0	Participant 1	71	130	12		4
0	Participant 2	72	41	3		3
0	Participant 3	52	87	1	Facility not available	2
0	Participant 4	81	120	18		4
0	Participant 5	47	100	3		3
0	Participant 6	116	189	19		4
1	Participant 7	51	69	3	0	4
1	Participant 8	74	89	4	12	4
1	Participant 9	41	98	6	4	3
1	Participant 10	91	135	14	0	4
1	Participant 11	58	66	13	7	4
1	Participant 12	58	57	6	3	4
2	Participant 13	68	102	13	12	4
2	Participant 14	71	159	3	0	3
2	Participant 15	63	118	12	6	4
2	Participant 16	50	34	5	5	4
2	Participant 17	21	10	2	1	3
2	Participant 18	49	74	14	0	4
3	Participant 19	84	104	4	2	3
3	Participant 20	55	105	5	7	4
3	Participant 21	67	85	6	2	3
3	Participant 22	39	48	6	5	4
3	Participant 23	17	41	4	0	2
3	Participant 24	57	127	6	1	4
4	Participant 25	41	51	1	0	2
4	Participant 26	70	67	22	6	4
4	Participant 27	35	65	3	10	3
4	Participant 28	66	80	8	0	4
4	Participant 29	65	100	4	11	3
4	Participant 30	61	65	6	12	4

Table D.4.1: Data collected for quantitative evaluation is tabulated in this table.

MLTutor Version	Mean	Time spent	Built-in links visited	Bookmarks visited	Suggestion list links visited	Web sites visited
0	Mean	73.2	111.1	9.3	Facility not available	3.3
	Stdev	24.6	49.2	8.1		0.8
1	Mean	62.2	85.7	7.7	4.3	3.8
	Stdev	17.8	28.6	4.7	4.6	0.4
2	Mean	53.7	82.8	8.2	4	3.7
	Stdev	18.4	55.1	5.4	4.7	0.5
3	Mean	53.2	85	5.2	2.8	3.3
	Stdev	23.1	34.2	1	2.6	0.8
4	Mean	56.3	71.3	7.3	6.5	3.3
	Stdev	14.6	16.8	7.6	5.4	0.8

Table D.4.2: Contains summary statistics for the data in table D.4.1.

Participant Number	MLTutor Version	2	5	12	81	82	83	74	77	67	66	94	95	101	104	109	110	112	114	115	116	118	121	122	123	128	127	126	129	130	132	133	Total link	Total visited pages
Participant 1	0						2	1				1						3	2	1	2	2	1		2	1		1				1	20	142
Participant 2	0						1	1				1						1		1		1			1								7	44
Participant 3	0	2		3		2	8	1												2			1										15	88
Participant 4	0	1		2		3	8	12	4	4		1						4	1	1	1	1	1		1	2	2	1					40	138
Participant 5	0	4	1				7	1														1	1		1			1					13	103
Participant 6	0					1	8	3				6						6	5	3	2	1	2		4	1								45	208
Participant 7	1			1			4	2				1						1	1		1	1	2		5								18	72
Participant 8	1	1		4				4	1	2								2	1	1		3			3								21	105
Participant 9	1		1	1	1	4	4	1										8	1	1		1											21	106
Participant 10	1						4	2		3		2			1			8	6	1	2	3			2								34	153
Participant 11	1						1	1				2						5	2	1	1	1	2			4		2				1	23	88
Participant 12	1							3				1						3	1		1				1	8	1	4	2	1			24	66
Participant 13	2						2	1				2						2		1		1			1	1		1		1	1	2	16	127
Participant 14	2		1			11	6	1				2																					24	162
Participant 15	2					1	3	3				2						8				2	3		3								23	138
Participant 16	2		2			1		1					1		2							2			2	1						1	13	44
Participant 17	2							1				1						1	1			1											5	13
Participant 18	2					1	2	1				2						4				1	2		1	2						1	17	66
Participant 19	3			1		2		15	1	1		1						3				1	1		2								28	110
Participant 20	3	1	1	3		14	8	1				2	1					3	1		1	2	2		2								36	117
Participant 21	3		1	2		1		1										6	1		1	2	1				2						18	83
Participant 22	3		2	2		1		2				1					1	1	1	1		1	1		1								15	48
Participant 23	3	1		2				1										1	2			1	1		3								11	45
Participant 24	3		1	1			1	2			3	4	2		1			1	1	1	1	1	1		1								22	134
Participant 25	4			2			8	2	1																								11	52
Participant 26	4					2	1	2				1						4		2	1	1	1		2								17	95
Participant 27	4			1														4	2	1		2		3	2								15	78
Participant 28	4							3				2						3	2	2		1			1	1			1	1		3	20	88
Participant 29	4		1	3	1		4	1										4	1	2		2			1								20	115
Participant 30	4					8	1	1	2			2						2	2			1	1										16	83

Table D.5.1: This table presents data showing frequency of solution page access via built-in hypertext links.

Participant Number	MLTutor Version	2	5	12	61	82	63	74	77	67	88	65	101	104	108	110	112	114	115	116	118	121	122	123	126	127	126	128	130	132	133	Total	Total visited pages
Participant 1	0									N	O	T						A	P	P	L	I	C	A	B	L	E					0	142
Participant 2	0																															0	44
Participant 3	0																															0	68
Participant 4	0																															0	138
Participant 5	0																															0	103
Participant 6	0																															0	208
Participant 7	1																															0	72
Participant 8	1	1			1						1	1						1														5	105
Participant 9	1				1		1														1											3	108
Participant 10	1																		2													2	153
Participant 11	1																															0	86
Participant 12	1																				1		1									2	68
Participant 13	2						1				1																		1			3	127
Participant 14	2																															0	182
Participant 15	2						2																									2	136
Participant 16	2				1		1												1													3	44
Participant 17	2																															0	13
Participant 18	2																															0	88
Participant 19	3						1																									1	110
Participant 20	3		1				2													2			1									8	117
Participant 21	3																			1												1	93
Participant 22	3				1		1											1								1						4	48
Participant 23	3																															0	45
Participant 24	3					1																										1	134
Participant 25	4																															0	52
Participant 26	4																		3	1	1											5	85
Participant 27	4																1	1	3	2			2	1								10	78
Participant 28	4																															0	68
Participant 29	4				1												2	1	1		1			1								7	115
Participant 30	4				1													1														2	83

Table D.5.2: This table presents data showing the frequency of solution page access via suggestion list links.

MLTutor Version	Participant Number	X1	X2	O1	O2
0	Participant 1	10		12	
0	Participant 2	9		3	
0	Participant 3	6	Not applicable	3	Not applicable
0	Participant 4	18		12	
0	Participant 5	8		1	
0	Participant 6	10		13	
1	Participant 7	13		2	
1	Participant 8	7	3	2	5
1	Participant 9	6		5	3
1	Participant 10	10	1	5	
1	Participant 11	15		6	
1	Participant 12	18		3	2
2	Participant 13	12	3	3	1
2	Participant 14	6		2	
2	Participant 15	7	2	3	
2	Participant 16	9		2	5
2	Participant 17	7		2	
2	Participant 18	10		7	
3	Participant 19	8		5	2
3	Participant 20	10	3	6	3
3	Participant 21	12	1	3	1
3	Participant 22	11		3	5
3	Participant 23	9		2	
3	Participant 24	11		10	2
4	Participant 25	3		1	
4	Participant 26	7	3	4	1
4	Participant 27		7	2	3
4	Participant 28	14		2	
4	Participant 29	3	8	2	2
4	Participant 30	6	2	7	3

Table D.6.1: This table shows solution page access routes in terms of task results using the scheme presented in Chapter 9.

MLTutor Version	Participant No	Mean	X1	X2	O1	O2
0	Participant 1					
0	Participant 7					
0	Participant 13	**Mean**	**10.2**	Not applicable	**7.3**	Not applicable
0	Participant 19	STDEV	4.1		5.5	
0	Participant 25					
0	Participant 2					
1	Participant 8					
1	Participant 14					
1	Participant 20	**Mean**	**11.5**	**2**	**3.8**	**3.3**
1	Participant 26	STDEV	4.7	1.4	1.7	1.5
1	Participant 3					
1	Participant 9					
2	Participant 15					
2	Participant 21					
2	Participant 27	**Mean**	**8.5**	**2.5**	**3.1**	**3**
2	Participant 4	STDEV	2.3	0.7	1.9	2.8
2	Participant 10					
2	Participant 16					
3	Participant 22					
3	Participant 28					
3	Participant 5	**Mean**	**10.2**	2	**4.8**	**2.6**
3	Participant 11	STDEV	1.5	1.4	3	1.5
3	Participant 17					
3	Participant 23					
4	Participant 29					
4	Participant 6					
4	Participant 12	**Mean**	**6.6**	**5**	**3**	**2.3**
4	Participant 18	STDEV	4.5	3	3.3	1
4	Participant 24					
4	Participant 30					

Table D.6.2: Contains summary statistics for the data in table D.6.1.

Participant Number	Q1	Q2	Q3	Q4	Q5	Q6
Participant 1	4	0	0	0	0	6
Participant 2	8	7	0	0	0	8
Participant 3	8	4	0	0	0	7
Participant 4	6	9	0	0	0	9
Participant 5	8	4	0	0	0	0
Participant 6	5	6	0	0	0	1
Participant 7	8	4	2	4	5	7
Participant 8	8	8	6	6	6	2
Participant 9	8	6	0	8	6	6
Participant 10	5	5	0	5	4	4
Participant 11	8	2	1	6	1	9
Participant 12	6	6	3	3	4	7
Participant 13	7	5	3	7	5	9
Participant 14	9	8	1	8	8	8
Participant 15	4	7	6	9	5	9
Participant 16	7	7	2	5	5	9
Participant 17	2.5	4.5	5.5	4.5	4.5	5.5
Participant 18	6	6	0	0	0	6
Participant 19	7	6	0	6	3	2
Participant 20	6	2	2	7	5	4
Participant 21	3	5	2	7	5	4
Participant 22	7	6	5	2	2	6
Participant 23	5.5	5.5	5.5	6.5	4.5	5.5
Participant 24	3	1	0	0	1	8
Participant 25	7	7	0	2	3	0
Participant 26	3	6	5	2	2	1
Participant 27	4	3	5	3	4	1
Participant 28	7	9	0	0	3	7
Participant 29	2.5	3.5	4.5	3.5	3.5	4.5
Participant 30	6	7	5	6	6	6

Table D.7.1: This table presents data from the user feedback questionnaire shown in C.3.

MLTutor Version	Mean	Q1	Q2	Q3	Q4	Q5	Q6
0	Mean	6.5	5	0	0	0	5.2
	Stdev	1.8	3	0	0	0	3.8
1	Mean	7.2	5.2	2	5.3	4.3	5.8
	Stdev	1.3	2	2.2	1.8	1.9	2.5
2	Mean	6	6.3	3	5.6	4.6	7.8
	Stdev	2.3	1.3	2.4	3.2	2.6	1.6
3	Mean	5.3	4.3	2.4	4.8	3.4	5
	Stdev	1.8	2.2	2.4	3	1.7	1.7
4	Mean	5	6	3.3	2.8	3.6	3.3
	Stdev	2	2.3	2.5	2	1.4	3

Table D.7.2: Contains summary statistics for the data in table D.7.1.

D.1 Users' Comments on MLTutor

Question 7; **What did you like about the MLTutor system?**

Version 0 (control)	G1:	Nothing.
	G2:	Bookmark system, it is very simple to use.
	G3:	Nothing, this was an ordinary Web-browsing system.
	G4:	It gave a good explanation of the processes going on the in the environment. It was relatively easy to use although it took a bit of time to familiarise myself with the system at the start.
	G5:	All encompassing Website, easy to understand.
	G6:	Easy to switch between topics and related links.
Version 1	G1:	Provides relevant information about environmental problems in an easy way to use and understand.
	G2:	The system is written in an easy language and the icons are useful to move on to different sites. There is a good coverage of up-to-date topics and waiting times on the WWW are eliminated.
	G3:	Clear design and good suggestions.
	G4:	It seems to be quite straightforward.
	G5:	A good collection of hypertext pages with all unnecessary links removed. The bookmarks were useful.
	G6:	Useful having links there permanently some suggestions helpful.
Version 2	G1:	The way it offered category headings on topics and the way it brought relevant and associated topics together.
	G2:	It is very easy to use and straight forward.
	G3:	Very user friendly easy to read.
	G4:	User friendly, plenty of options of links.
	G5:	-
	G6:	Website bookmarks.
Version 3	G1:	Lots of useful information.
	G2:	It gives the information in a concise way, so it is easier to find what you oeed.

G3: -

G4: I like the bookmarks at the bottom of the screen and suggestion sometimes.

G5: An additional choice.

G6: Bookmarks.

Version 4

G1: Easy to navigate and find your way around.

G2: Colourful and, figures and charts are easy to understand.

G3: Some suggestions were good and provided a 'shortcut' to a page that was hard
to access immediately.

G4: The system was easy to use and was contained relevant information.

G5: It was helpful in having different forms of navigational aids.

G6: Made me aware of links I did not know about.

Question 8: What didn't you like about the system?

Version 0 (control)

G1: It didn't feel like a 'tutor' system, it has no structure. Sites can be very lacking in information; there is no way of telling if you are on the right trade.

G2: -

G3: Nothing.

G4: The different web pages (4) in total could be have been linked together where the same topic was being discussed e.g. ground level ozone production was on two different web sites and the two web sites were not linked to one another. There was a lot of reading needed.

G5: The screen size and I had to do lots of scrolling.

G6: Cannot use the search function.

Version 1

G1: There should be more Web pages in the system and useful suggestions.

G2: I liked the set up of the ozone depletion page better than the other Web pages.
Because all the topics are given on the first page, so it is easier to look them up.

G3: List of suggestions is too long and categories mixed.

G4: Suggestions seemed to cover mostly pages, which already have been visited.

	G5:	The suggestion box was behind the main window and it was not obvious for any use.
	G6:	The format of display of suggestions not very intuitive. Can be quite some time after doing something that a useful suggestion appears.
Version 2	G1:	Many pages with little information had priority with MLTutor, especially if visited first. The topic menu disappears after clicking on one topic.
	G2:	I don't know if relevant: Some of the topics are presented in more than one page and this makes more difficult to find the right answer. However this is what the Internet is all about.
	G3:	A 'layman' needs to access some areas before finding out if they are useful.
	G4:	The entries in the suggestion list were very difficult to understand. Furthermore, there was no explicit content structure.
	G5:	The suggestion box was behind the main window – it wasn't obvious it was for my use.
	G6:	The list of the suggestions was too long and the categories in the suggestion list were mixed.
Version 3	G1:	The font, colours and general effects.
	G2:	Sometimes I wanted to use a suggestion and it was deleted before I tried.
	G3:	-
	G4:	Suggestions were not available immediately when I wanted them. I felt restricted with the number of pages I could visit.
	G5:	The choices were not very indicative.
	G6:	Suggestion box – didn't seem to work.
Version 4	G1:	Some of the information is scientifically incorrect although as far as getting the information access to the general public is most probably effective.
	G2:	Limitation of details in each topic and hard to find relevant information.
	G3:	The hints were vague and repetitive.
	G4:	Some links, which I thought might have been useful, were disabled. However this could lead to 'blind alleys '.
	G5:	Some speed problems, I thought that it would made searching easier if the MLTutor alphabetically indexed instead of topic indexed. I thought the wording for finding certain topics was not precise enough.
	G6:	Window was a bit small I had to scroll for links.

Appendix D

Question 9: **Could you suggest any improvements?**

Version 0 (control)	G1:	Questionnaire could be on screen and interactive (multiple choice or reveal answers).
	G2:	Ability to add to bookmarks.
	G3:	None.
	G4:	More diagrams would be good. Also pictures of the effects of acid rain, ozone depletion and ground level ozone production and the photographs of photochemical smog and linking the four web sites together somehow.
	G5:	A side menu bar for links, a site map, a bigger glossary and a help page.
	G6:	Keyword search.
Version 1	G1:	More web pages and an optional language.
	G2:	-
	G3:	Better categories and more combined with a search engine.
	G4:	No.
	G5:	Bookmarks could be arranged according to subjects rather than the original sites the pages were taken from.
	G6:	Some way of displaying suggestions more intuitively –less obscure textual format.
Version 2	G1:	Keep the topic window up at all times after 10 pages analysis completed. One topic i.e. smog may have more than one entry show all entries.
	G2:	-
	G3:	A 'tree' system to link topics.
	G4:	-
	G5:	I would like to have the option to ask questions about the web content and interact more closely with the system.
	G6:	The suggestion box on top.
Version 3	G1:	-
	G2:	-
	G3:	-
	G4:	A button on the MLTutor page to bring up suggestions when I wanted them and more bookmarks would be good.

G5: -

G6: Fix suggestion box make it adaptable in size.

Version 4

G1: Some of the pages especially those on acid rain could be linked in a more systematic and progressive manner starting with general outlines etc. and becoming more in depth.

G2: Suggestions should be related to the topics.

G3: I would like a more specific selection of keywords.

G6: -

G5: Some type of graphical representation between navigational aids was needed, so the user can establish a relationship.

G6: Make a graphical representation i.e. branching flow chart.

Question 10: **Any other comments?**

Version 0 (control)

G1: -

G2: A guidance program was needed.

G3: -

G4: There was too much writing to do, in the small space available. A series of tick boxes would have been much better or at least some way to cut down on all the writing and direct copying of the text on the screen to the paper.

G5: -

G6: Try to be a more friendly to the users.

Version 1

G1: -

G2: -

G3: Experience computer users may find this less useful as have search habits. Children would probably like it.

G4: -

G5: I would much prefer a search engine to the system suggestions.

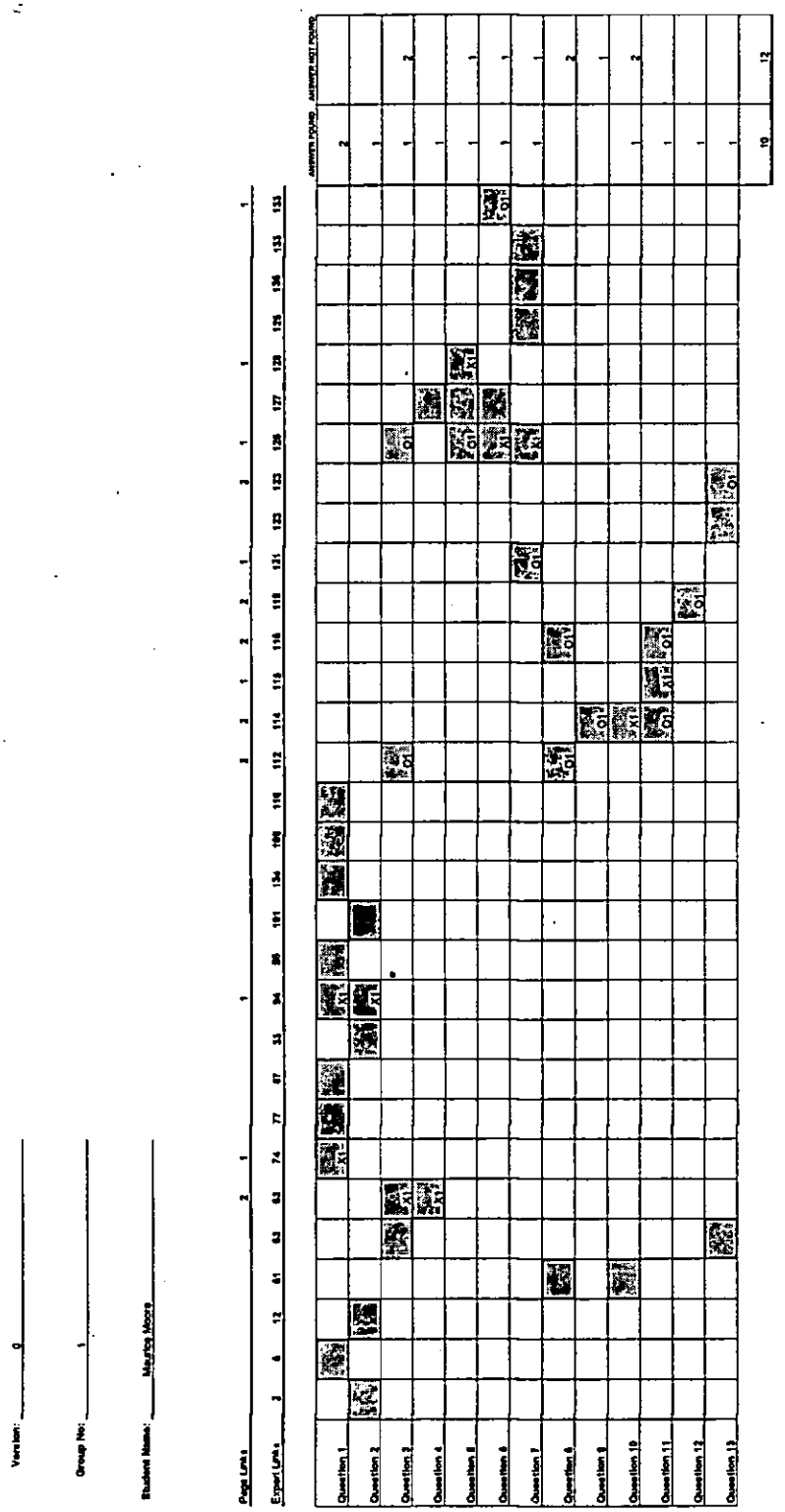
G6: -

Version 2

G1: Large amount of text on web pages without adequate sub-listing made searching information harder. The environmental problems main page was the most useful and constructively structured. More menu options with topic headings (especially not too vague headings) would help and save deep

		down searches of web pages. Reverse suggestion list window and topic window, so topic window brings up page/file names.
	G2:	-
	G3:	-
	G4:	Enjoyable to browse through.
	G5:	-
	G6:	The chosen Web sites were not easy to read (not the fault of MLTutor though). Some links were not available, links previously visited did not change colour –so you did not know where you had been previously – made repeated attempts to visit a site which was not available.
Version 3	G1:	-
	G2:	-
	G3:	-
	G4:	Good design.
	G5:	-
	G6:	-
Version 4	G1:	-
	G2:	-
	G3:	The set up feels a bit unnatural to me. I would rather print out the pages and read them and use a browser using keywords such as 'ground level ozone'. Also the print was very small, which was irritating and straining.
	G4:	-
	G5:	-
	G6:	It was an interesting experience, a useful project although I feel that the expert systems are teacher-centred rather than student-centred in their approach to knowledge acquisition.

D.2 Users' *link usage analysis* forms



Version: 0

Group No: 2

Student Name: TIM FLANNERY

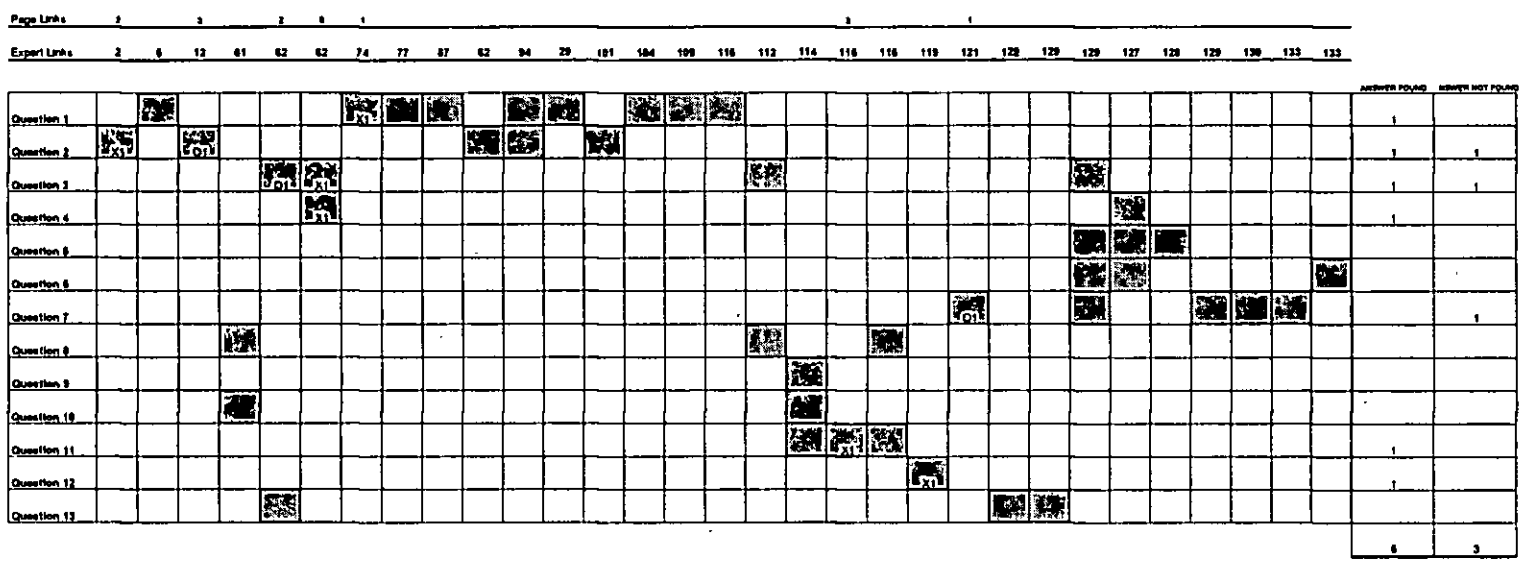
Page Links																														
Export Links	2	5	12	61	62	63	74	77	87	88	89	95	101	104	106	110	112	114	115	116	118	121	123	133	125	127	128	129	130	133	133

																														ANSWER FOUND	ANSWER NOT FOUND
Question 1							X1				X1																			2	
Question 2											X1																			1	
Question 3						X1											Q1													1	1
Question 4						X1																				Q1				1	1
Question 6																															
Question 8																															
Question 7																															
Question 8																	X1													1	
Question 9																															
Question 10																															
Question 11																		X1	Q1											1	1
Question 12																				C X1										1	
Question 13																								X1						1	
																														9	3

Version: 0

Group No: 3

Student Name: OLAF SCHMELT



Version: 0

Group No: 4

Student Name: PAUL OLIPPN

Page Links 1 3 5 8 12 4 4 1 4 1 1 1 1 1 1 2 3 1

Expert Links 2 9 12 61 64 94 74 77 87 69 94 98 101 104 108 110 112 114 118 118 119 121 122 122 126 127 128 129 130 133 153

																															ANSWER FOUND	ANSWER NOT FOUND
Question 1							X1	O1	O1		X1																				2	2
Question 2	X1		X1								X1																				3	
Question 3					X1	X1											O1								O1						2	2
Question 4						X1																				O1					1	1
Question 5																									X1	X1	O1				2	1
Question 6																									X1	O1				X1	1	1
Question 7																					O1				X1						1	1
Question 8				X1													O1			X1											1	1
Question 9																		O1														1
Question 10				X1														X1													1	
Question 11																		X1	X1	O1											2	1
Question 12																					X1											
Question 13					X1																		X1	O1							1	1
																															18	12

Version: 0

Group No: 5

Student Name: PAUL O'LEARY

Page Links	4	1				7	1																1	3		1			7							
Expert Links	2	8	12	61	62	82	74	77	87	92	94	98	101	104	109	110	113	114	116	119	118	121	122	123	126	127	128	129	136	132	133		

																																ANSWER FOUND	ANSWER NOT FOUND
Question 1		X1					X1																									2	
Question 2	X1																															1	
Question 3						X1																										1	
Question 4						X1																										1	
Question 5																											X1					1	
Question 6																																	
Question 7																																	1
Question 8																																	
Question 9																																	
Question 10																																	
Question 11																																	
Question 12																																1	
Question 13																																1	
																																6	1

Appendix D

Version: 0

Group No: 0

Student Name: QINGGANG MENG

Expert Links	3	6	12	61	62	63	74	77	87	88	94	96	101	104	106	110	112	114	115	116	118	121	123	125	126	127	128	129	130	132	133	ANSWERS FOUND	ANSWERS NOT FOUND
Page Links					1	3	1				8						8	1	1	2	1	2		4	1								
Question 1		■					X1	■	■		X1	■		■	■	■																2	
Question 2	■		■							■	O1		X1																			1	1
Question 3					O1	X1											X1								O1							2	2
Question 4						O1																				■						1	
Question 5																									X1	■	■					1	
Question 6																									O1	■					■		1
Question 7																						O1			O1			■	■	■			2
Question 8				■													X1			O1												1	1
Question 9																		O1															1
Question 10				■														X1														1	
Question 11																		X1	O1	O1												1	2
Question 12																					X1											1	
Question 13					O1																		■	O1									2
																																10	13

Version: 1

Group No: 1

Student Name: YOLANDA JIMENEN

Suggest Links

Page Links 1 4 3 1 1 1 1 2 5

Expert Links 2 6 12 61 62 63 74 77 87 23 94 96 101 104 109 110 112 114 116 118 119 121 122 123 126 127 128 129 130 132 133

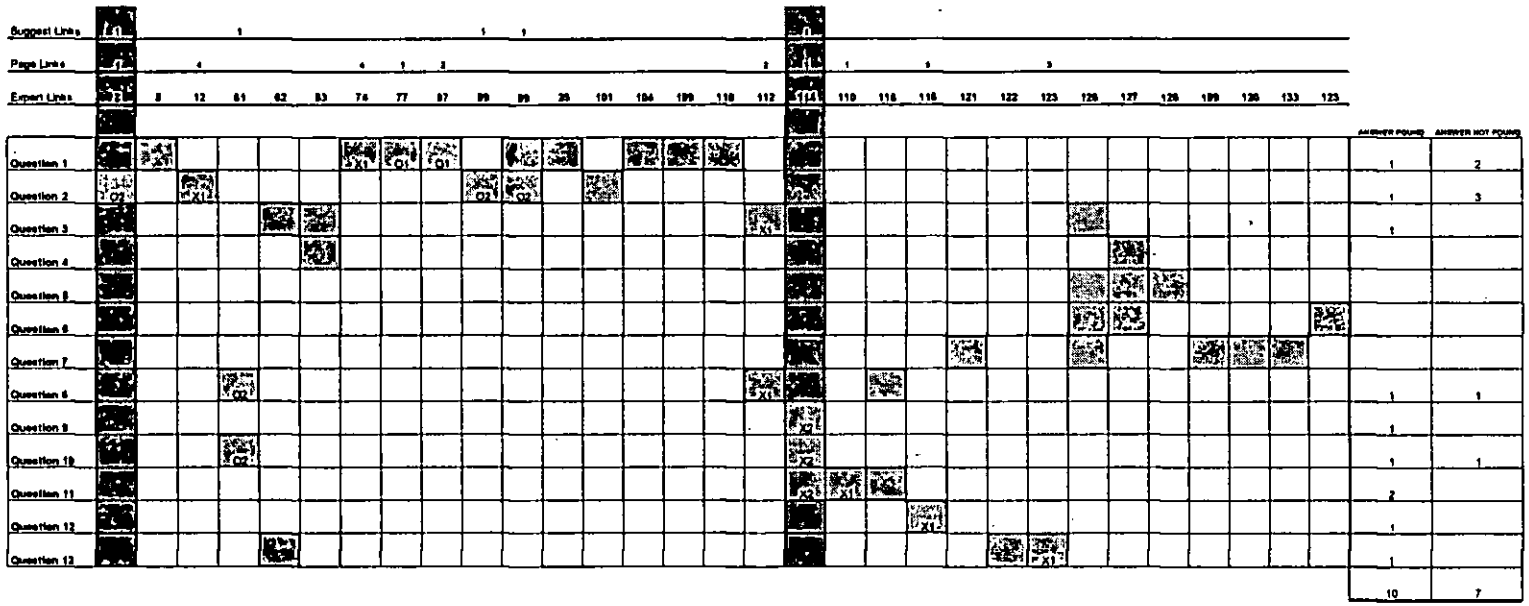
																																ANSWER FOUND	ANSWER NOT FOUND
Question 1							X1				X1																					2	
Question 2			X1								X1																					2	
Question 3						X1												Q1														1	1
Question 4						X1																										1	
Question 5																																	
Question 6																																	
Question 7																																1	
Question 8				X1														X1		Q1												1	1
Question 9																			X1													1	
Question 10				X1															X1													1	
Question 11																			X1		X1											2	
Question 12																					X1											1	
Question 13					X1																		X1									1	
																																13	2

Appendix D

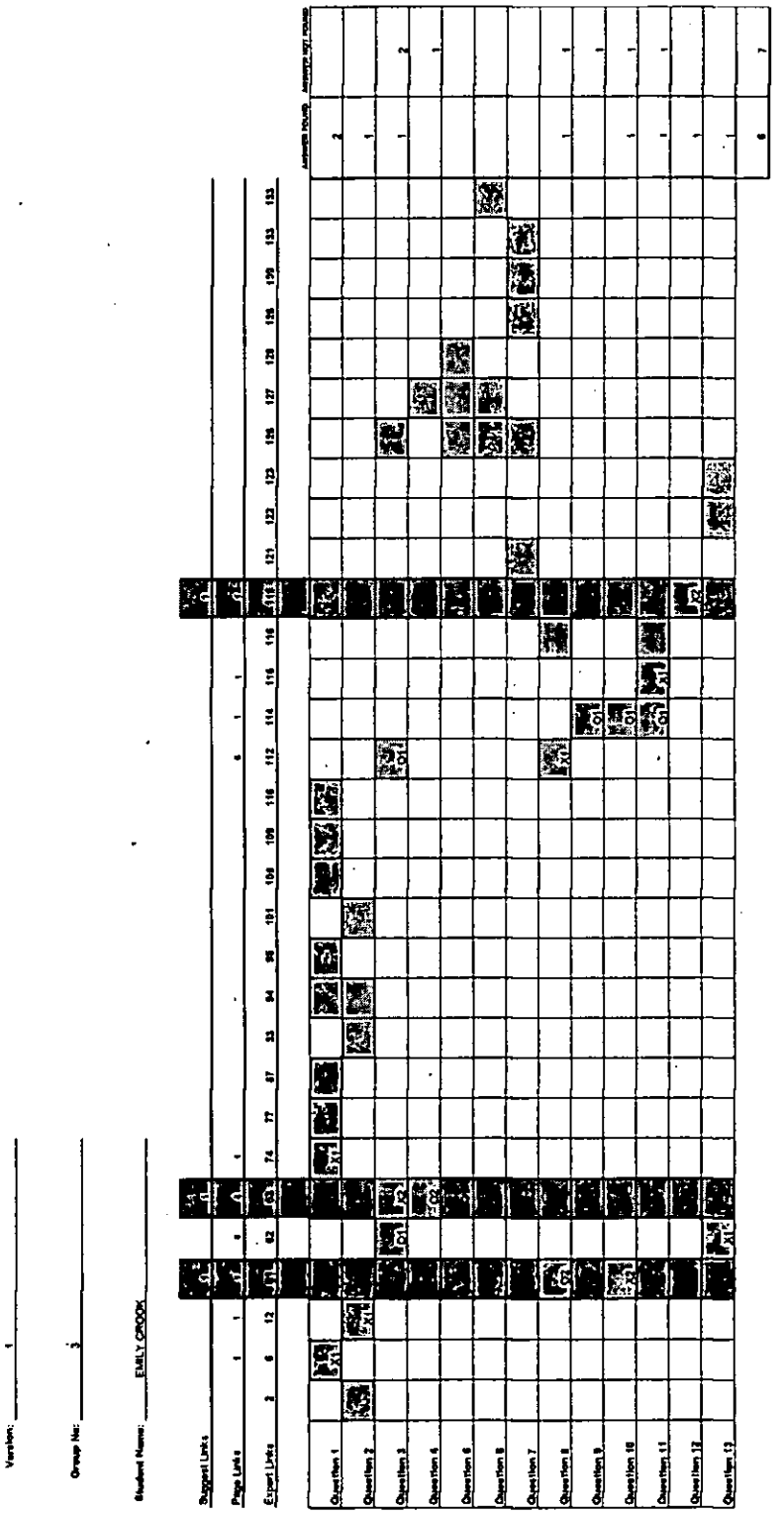
Version: 1

Group No: 2

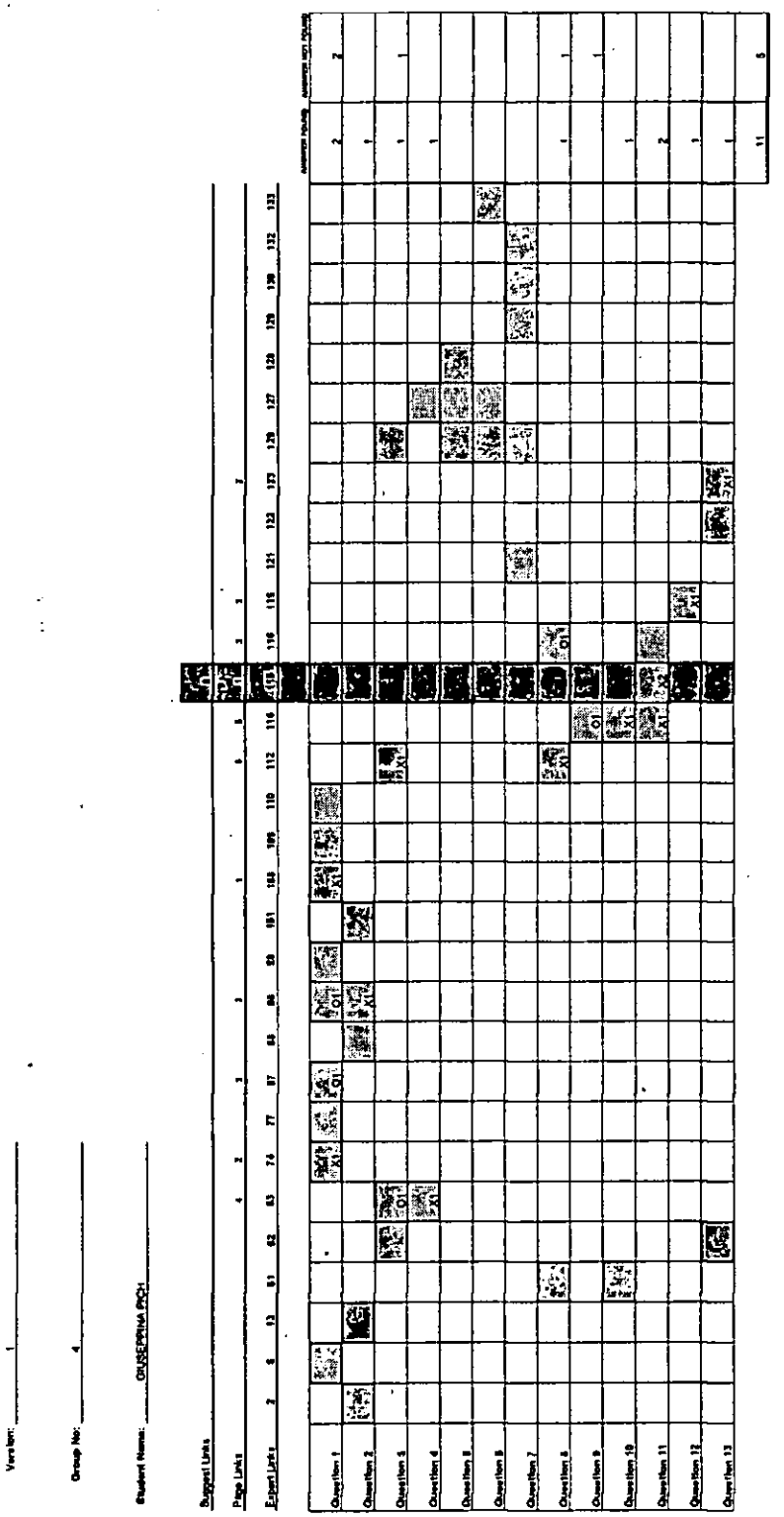
Student Name: Judith Roether



Appendix D


Version: 1
Group No: 3
Student Name: EMILY CROOK
Suggest Links
Page Links
Expert Links
Question 1
Question 2
Question 3
Question 4
Question 5
Question 6
Question 7
Question 8
Question 9
Question 10
Question 11
Question 12
Question 13

Appendix D



Appendix D

Version: 1

Group No: 5

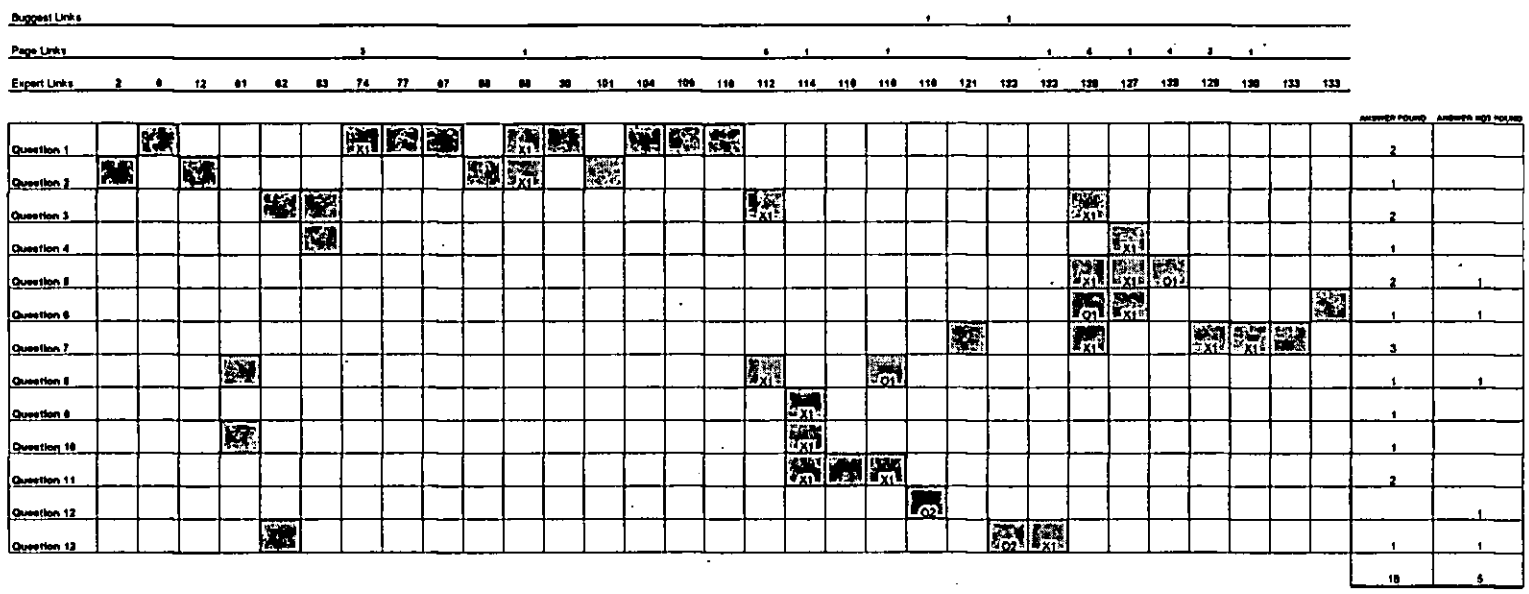
Student Name: CASPER NIELSEN

																																ANSWER FOUND	ANSWER NOT FOUND
Suggest Links																																	
Page Links						1	1		·		3						6	3	1	1	1	2			4		3				1		
Expert Links	2	9	13	61	63	66	74	77	87	88	94	96	101	104	109	110	112	114	116	118	119	121	123	123	126	127	128	129	130	123	133		
Question 1		■					X1	■	■		X1	■		■	■	■																2	
Question 2	■		■							■	X1		■																			1	
Question 3					■	X1											O1								O1							1	2
Question 4						X1																				■						1	
Question 5																									X1	■	■					1	
Question 6																									X1	■					X1	2	
Question 7																						X1			O1			O1	■	■		1	2
Question 8				■													X1			O1												1	1
Question 9																		X1														1	
Question 10				■														X1														1	
Question 11																		O1	X1	X1												2	1
Question 12																					X1											1	
Question 13					■																		■	■									
																																15	6

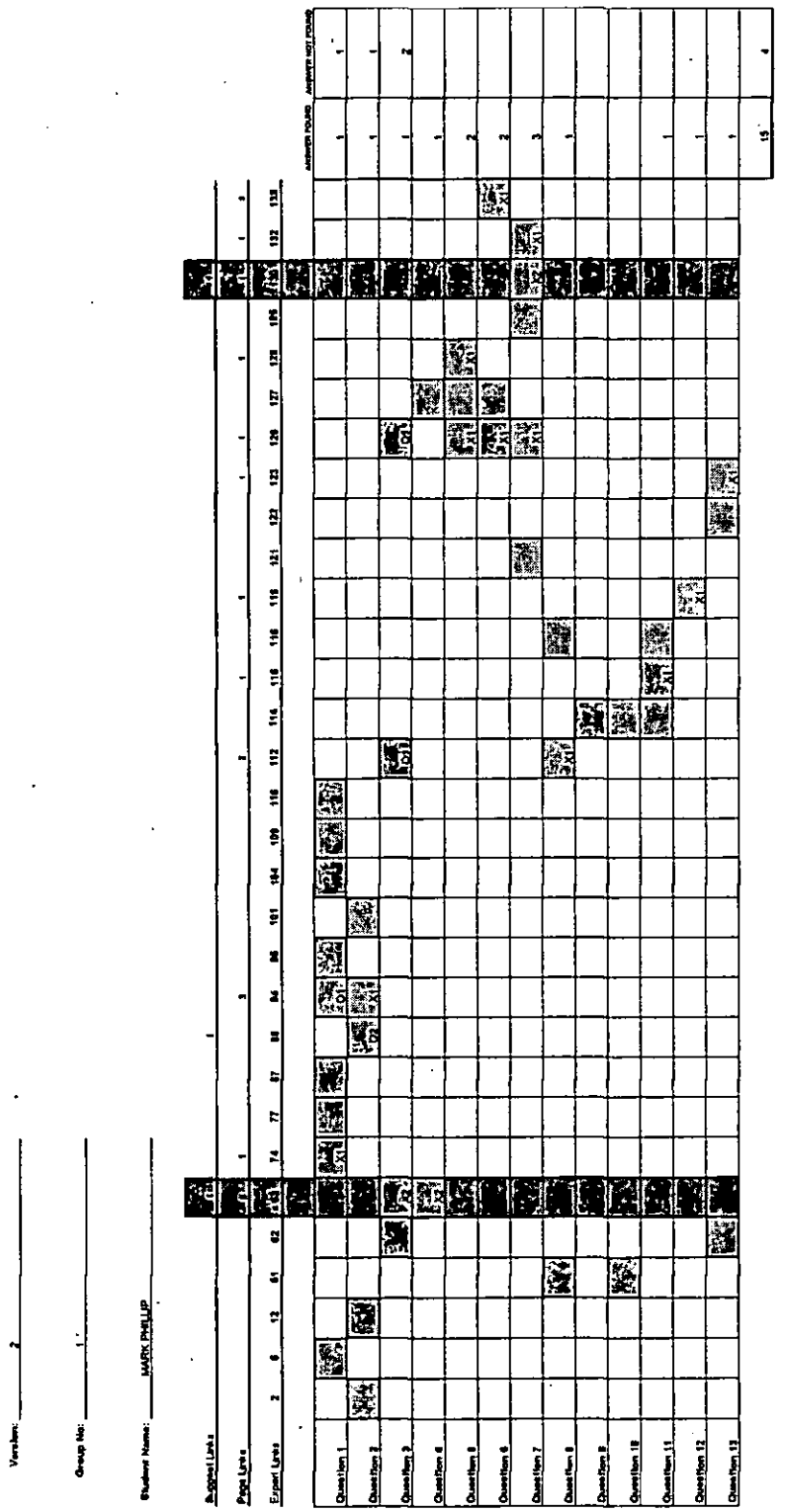
Version: 1

Group No: 6

Student Name: PETER STEVENS



Appendix D


Version: 2
Group No: 1
Student Name: MARK PHILLIP
Suggest Links
Page Links
Expert Links
Question 1
Question 2
Question 3
Question 4
Question 5
Question 6
Question 7
Question 8
Question 9
Question 10
Question 11
Question 12
Question 13
ANSWER FOUND
ANSWER NOT FOUND

Version: 2

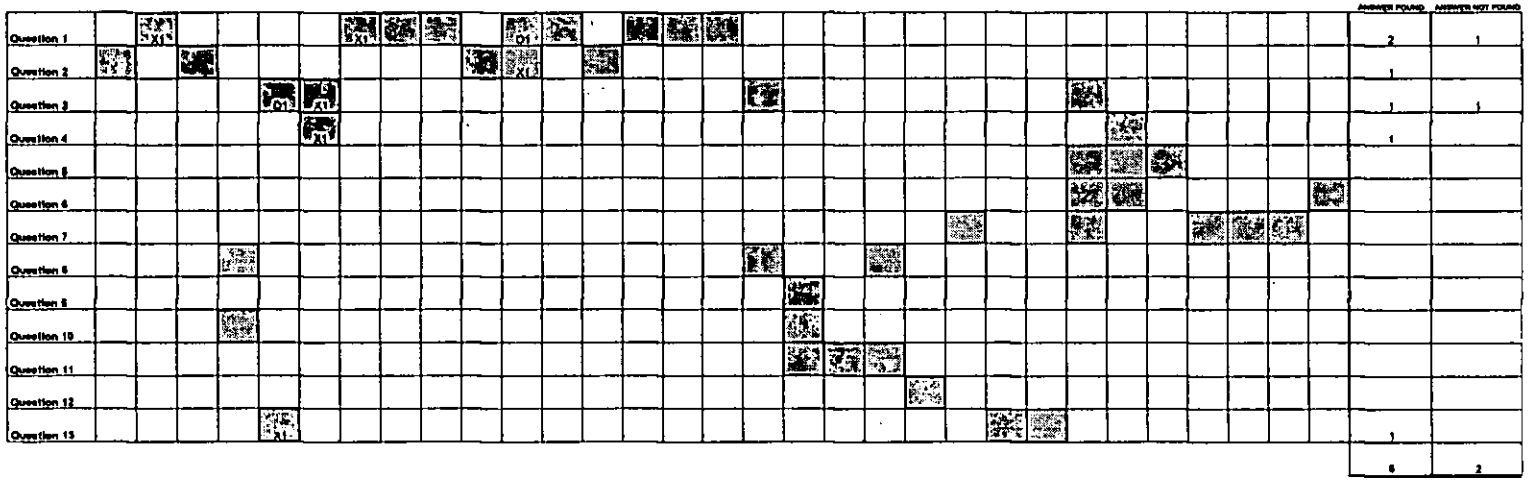
Group No: 2

Student Name: GIANLUCA BENATI

Suggest Links

Page Links 1 11 6 1 2

Expert Links 2 8 13 61 62 63 74 77 87 88 90 96 101 104 109 110 112 114 116 116 116 118 121 123 123 126 127 128 129 130 133 133

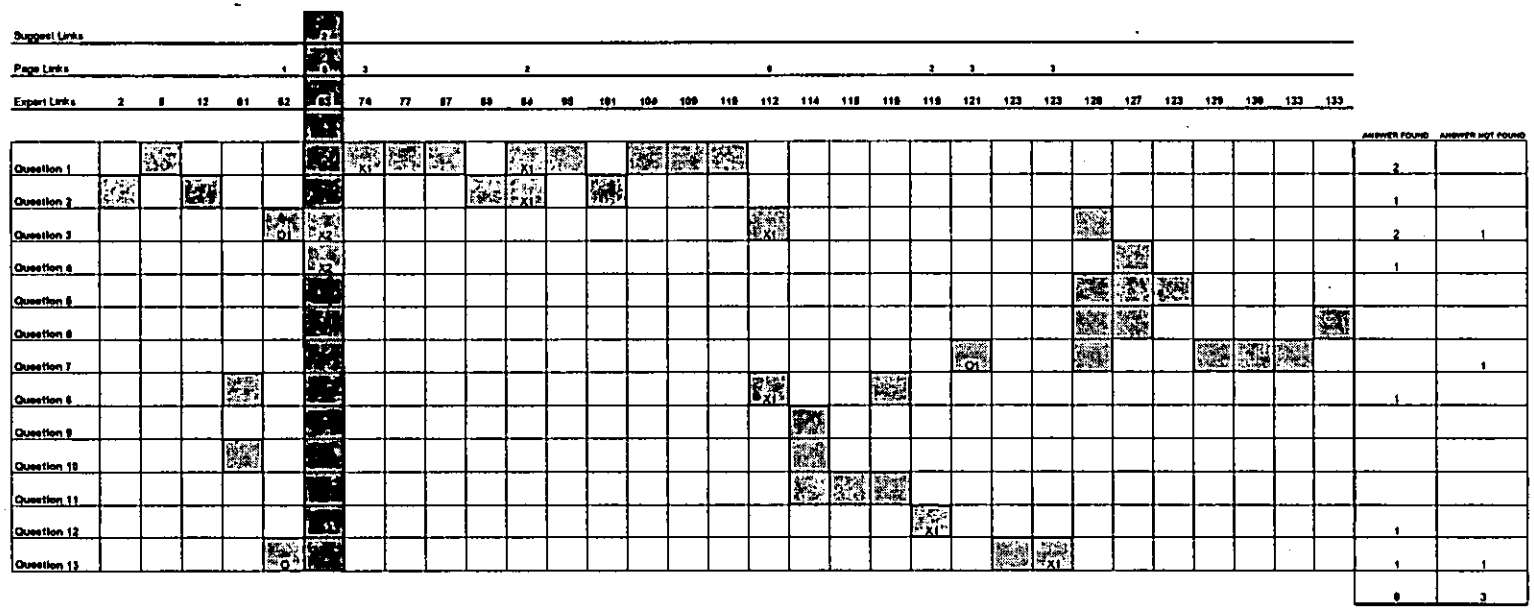


	ANSWER FOUND	ANSWER NOT FOUND
Question 1	2	1
Question 2	1	
Question 3	1	1
Question 4	1	
Question 5		
Question 6		
Question 7		
Question 8		
Question 9		
Question 10		
Question 11		
Question 12		
Question 13	1	
	6	2

Version: 2

Group No: 3

Student Name: MARTIN BOPER



Appendix D

Version: 2

Group No: 4

Student Name: LIAN SCHORES

	3	6	13	61	62	63	74	77	87	88	94	96	101	104
Suggest Links				1		1								
Page Links		2			1		1					1		2
Expert Links	3	6	13	61	62	63	74	77	87	88	94	96	101	104
Question 1		X1					X1	■	■		■	O1		X1
Question 2	■		■							■	■		■	
Question 3					X1	O2								
Question 4						O2								
Question 5														
Question 6														
Question 7														
Question 8				O2										
Question 9														
Question 10				O2										
Question 11														
Question 12														
Question 13					O1									

	109	110	112	114	115	116	119	121	122	123	126	127	128	129	130	132	133	ANSWER FOUND	ANSWER NOT FOUND
Suggest Links					1														
Page Links							2			2	1						1		
Expert Links	109	110	112	114	115	116	119	121	122	123	126	127	128	129	130	132	133		
Question 1	■	■																3	1
Question 2																			
Question 3			■								■							1	1
Question 4												■							1
Question 5											X1	■	■					1	
Question 6											X1	■					■	1	
Question 7								■			X1			■	■	■		1	
Question 8			■			■													1
Question 9				■															
Question 10				■															1
Question 11				■	O2	■													1
Question 12							X1											1	
Question 13									■	X1								1	1
																		6	7

Version: 2

Group No: 5

Student Name: KOSTAS GIANNAKIS

Expert Links	3	8	12	81	82	83	76	77	87	90	94	98	101	104	104	110	112	114	118	118	118	121	123	123	126	127	128	129	130	132	133	ANSWER FOUND	ANSWER NOT FOUND
Suggest Links																																	
Page Links							1				1						1	1			1												
Question 1		■					■ X1	■	■		■ O1	■		■	■	■																1	1
Question 2	■		■							■	■ O1		■																				1
Question 3					■	■											■ X1								■							1	
Question 4						■																				■							
Question 5																									■	■	■						
Question 6																									■	■					■		
Question 7																						■			■			■	■	■			
Question 8				■													■ X1			■												1	
Question 9																		■ X1														1	
Question 10				■														■ X1														1	
Question 11																		■ X1	■	■												1	
Question 12																					■ X1											1	
Question 13					■																		■	■									
																																7	2

Version: 2

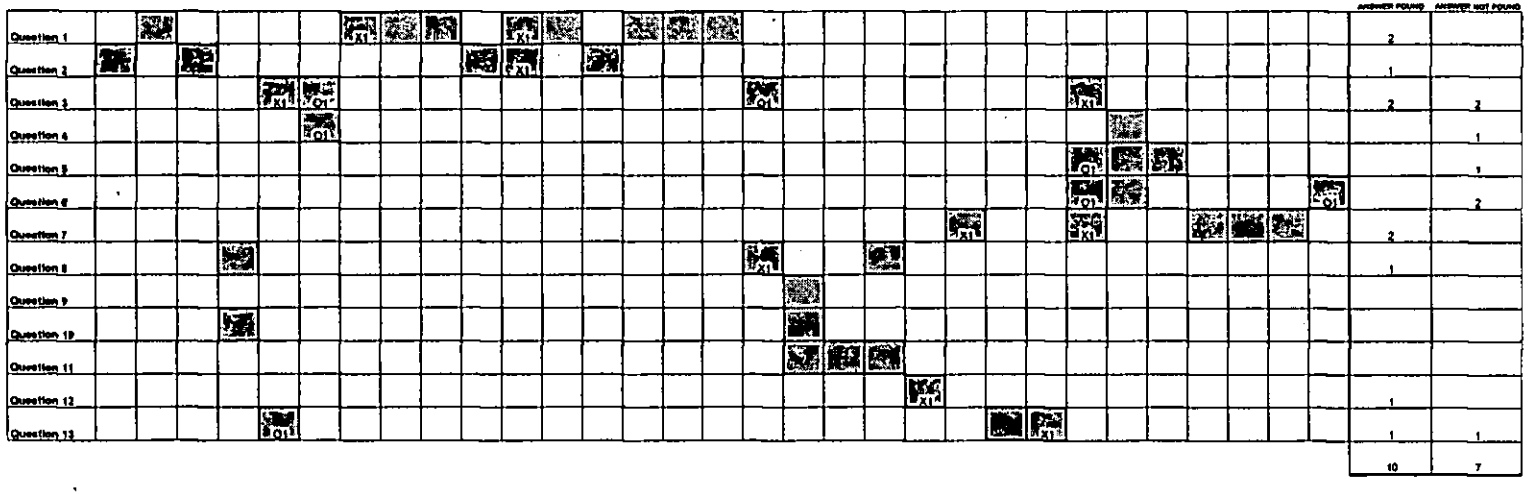
Group No: 6

Student Name: PENNY DUQUENOY

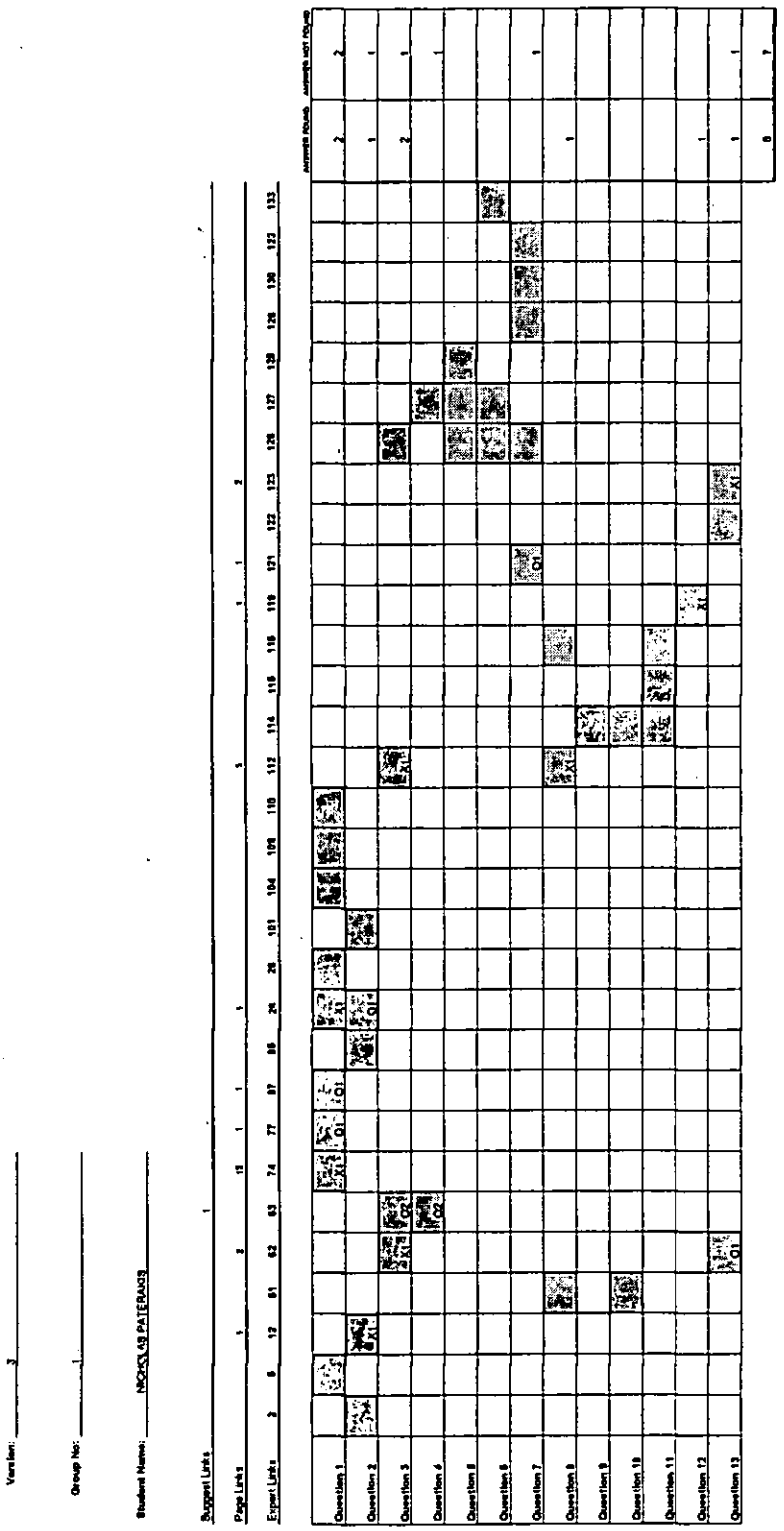
Suggest Links

Page Links 1 2 1 3 4 1 2 1 2 3

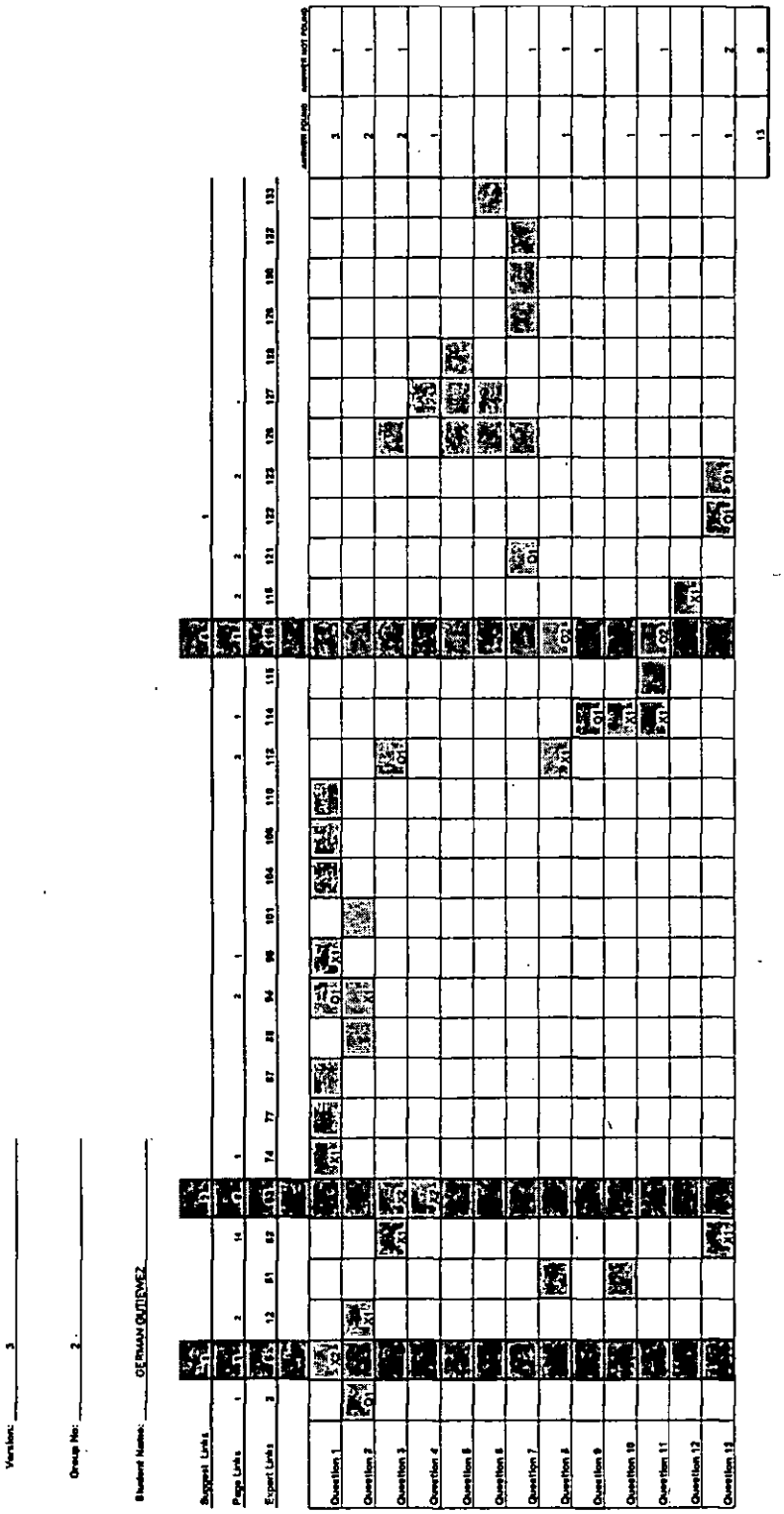
Expert Links 3 6 12 61 62 63 76 77 87 88 94 96 101 102 109 110 112 114 115 116 118 121 123 125 128 127 129 129 130 133 135



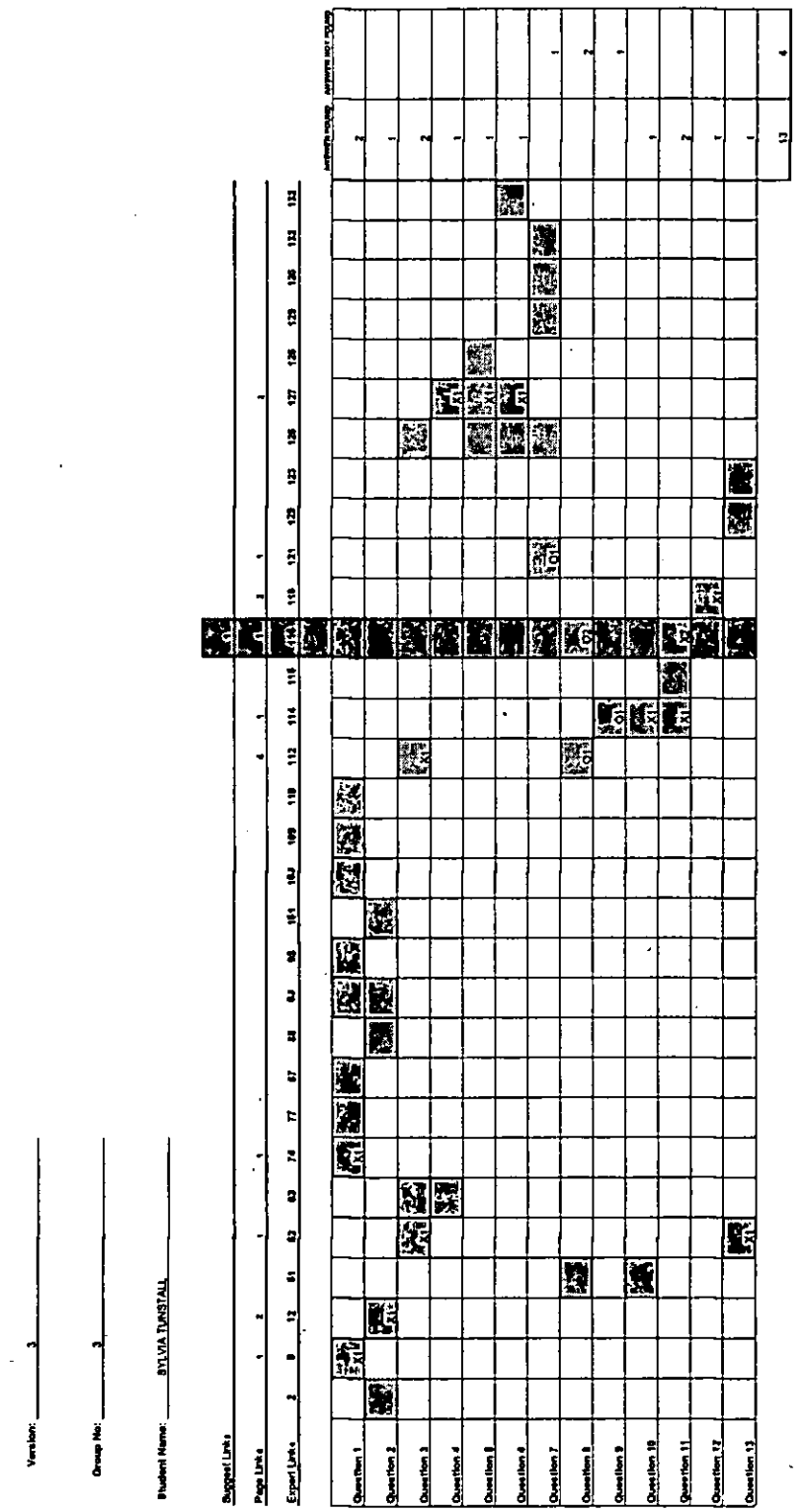
Appendix D


Version: 3
Group No: 1
Student Name: NICHOLAS PATERAKIS
Suggest Links
Page Links
Expert Links
Question 1
Question 2
Question 3
Question 4
Question 5
Question 6
Question 7
Question 8
Question 9
Question 10
Question 11
Question 12
Question 13

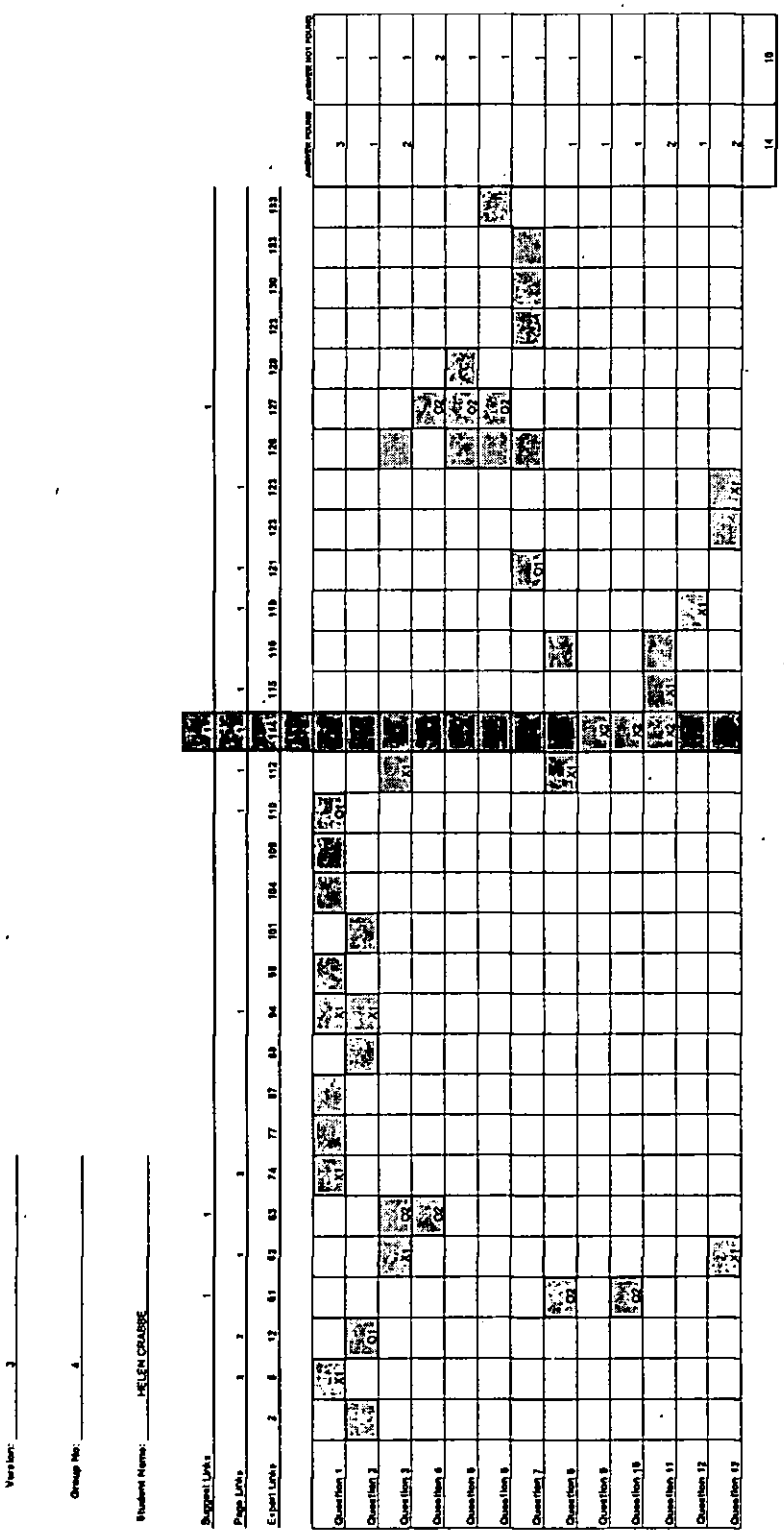
Appendix D


Version: 3
Group No: 2
Student Name: GERMAN GUTIERREZ
Biggest Links
Page Links
Expert Links
Question 1
Question 2
Question 3
Question 4
Question 5
Question 6
Question 7
Question 8
Question 9
Question 10
Question 11
Question 12
Question 13

Appendix D


Version: 3
Group No: 3
Student Name: SYLVIA TUNSTALL
Suggest Links
Page Links
Expert Links
Question 1
Question 2
Question 3
Question 4
Question 5
Question 6
Question 7
Question 8
Question 9
Question 10
Question 11
Question 12
Question 13

Appendix D


Version: 3
Group No: 4
Student Name: HELEN CRABBE
Suggest Links
Page Links
Expert Links
Question 1
Question 2
Question 3
Question 4
Question 5
Question 6
Question 7
Question 8
Question 9
Question 10
Question 11
Question 12
Question 13

Version: 3

Group No: 5

Student Name: THOMAS TAN

Suggest Links

Page Links 1 2 1 1 3 1 3 3

Expert Links 2 5 12 61 63 64 74 77 87 33 94 96 101 104 106 116 112 114 116 116 115 121 123 123 126 127 129 129 136 133 133

																																				ANSWER FOUND	ANSWER NOT FOUND
Question 1																																				1	
Question 2																																				2	
Question 3																																				1	
Question 4																																					
Question 5																																					
Question 6																																					
Question 7																																					1
Question 8																																				1	
Question 9																																				1	
Question 10																																				1	
Question 11																																				1	
Question 12																																				1	
Question 13																																				1	1
																																			10	2

Version: 3

Group No: 6

Student Name: JO HYDE

Suggest Links	1
Page Links	1 1 1 3 3 6 2 1 1 1 1 1 1 1 1 1 1
Expert Links	2 6 12 61 62 62 74 77 67 23 94 96 161 194 195 110 112 114 115 116 116 121 122 122 126 127 128 129 130 132 133

	ANSWER FOUND	ANSWER NOT FOUND
Question 1	2	3
Question 2	1	2
Question 3	1	2
Question 4	1	
Question 5		
Question 6		
Question 7		1
Question 8	1	1
Question 9		1
Question 10		1
Question 11	3	
Question 12	1	
Question 13	1	1
	11	12

Appendix D

Version: 4

Group No: 1

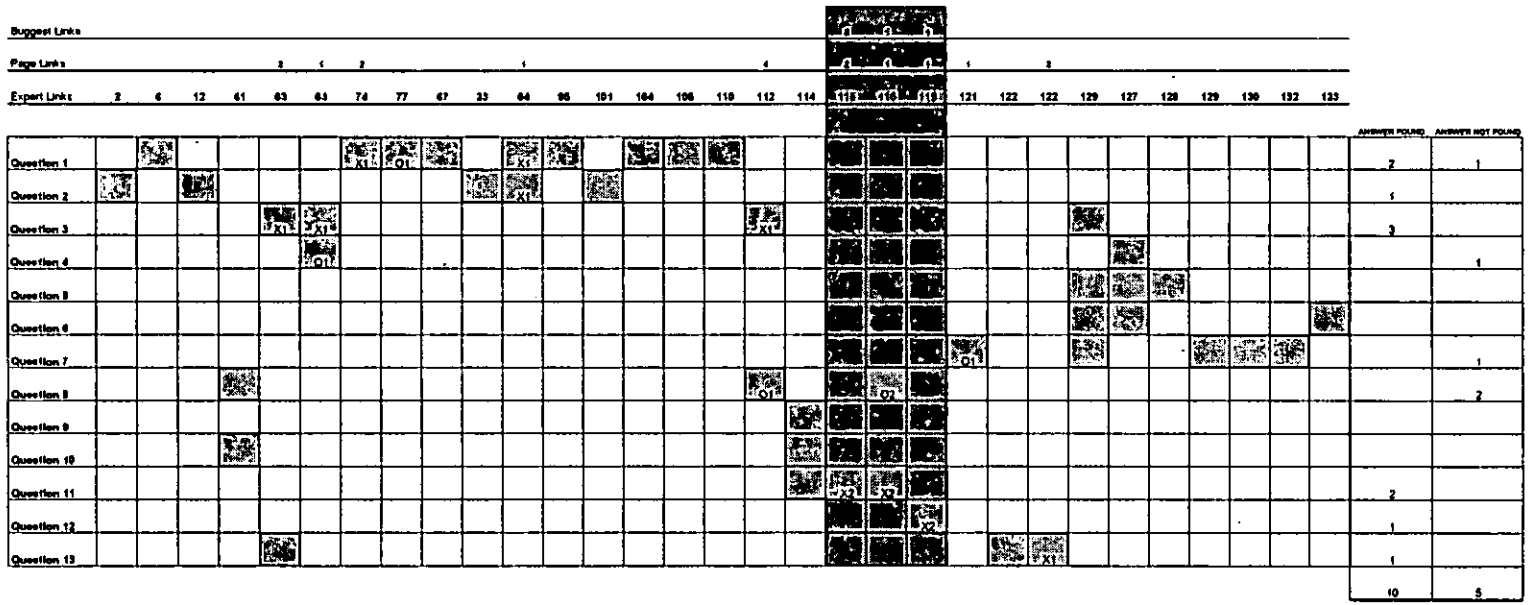
Student Name: DALE FISHER

Suggest Links																																
Page Links			3			6	2	1																								
Expert Links	2	8	12	61	62	63	74	77	87	88	94	95	101	104	109	110	112	114	118	116	119	121	123	123	126	127	128	185	186	184	183	ANSWER FOUND	ANSWER NOT FOUND
Question 1		[illegible]					[illegible]	[illegible]	[illegible]		[illegible]	[illegible]		[illegible]	[illegible]	[illegible]																1	1
Question 2	[illegible]		[illegible]							[illegible]	[illegible]		[illegible]																			1	
Question 3					[illegible]	[illegible]											[illegible]								[illegible]							1	
Question 4						[illegible]																				[illegible]						1	
Question 5																									[illegible]	[illegible]	[illegible]						
Question 6																									[illegible]	[illegible]					[illegible]		
Question 7																						[illegible]			[illegible]			[illegible]	[illegible]	[illegible]			
Question 8				[illegible]													[illegible]			[illegible]													
Question 9																		[illegible]															
Question 10				[illegible]														[illegible]															
Question 11																		[illegible]	[illegible]	[illegible]													
Question 12																					[illegible]												
Question 13					[illegible]																		[illegible]	[illegible]									
																																4	1

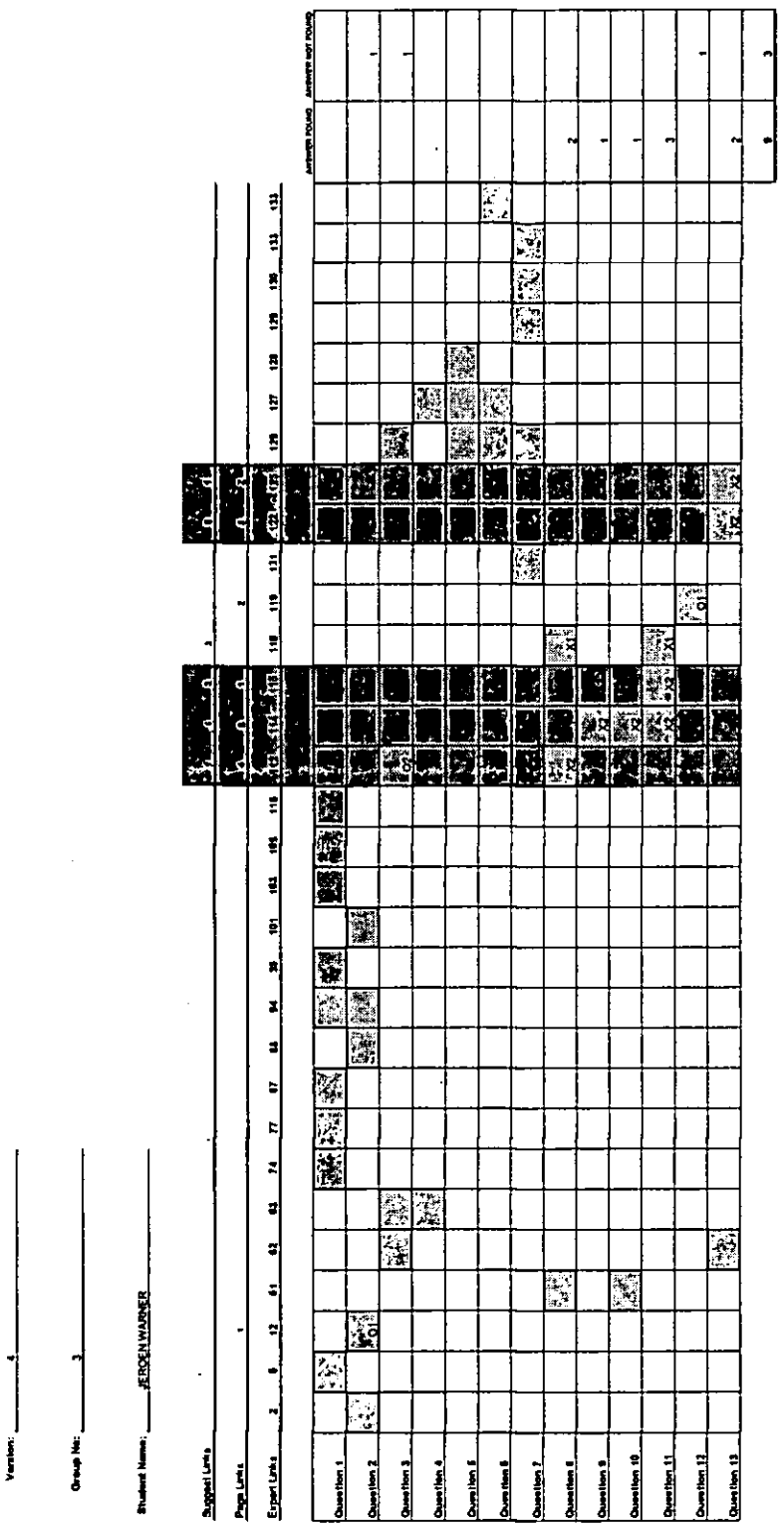
Version: 4

Group No: 2

Student Name: TEERAMET TITSEESANG



Appendix D


Version: 4
Group No: 3
Student Name: JEROEN WARNER
Suggest Links
Page Links
Expert Links
Question 1
Question 2
Question 3
Question 4
Question 5
Question 6
Question 7
Question 8
Question 9
Question 10
Question 11
Question 12
Question 13
ANSWERS FOUND
ANSWERS NOT FOUND

Version: 4

Group No: 4

Student Name: HUW JONES

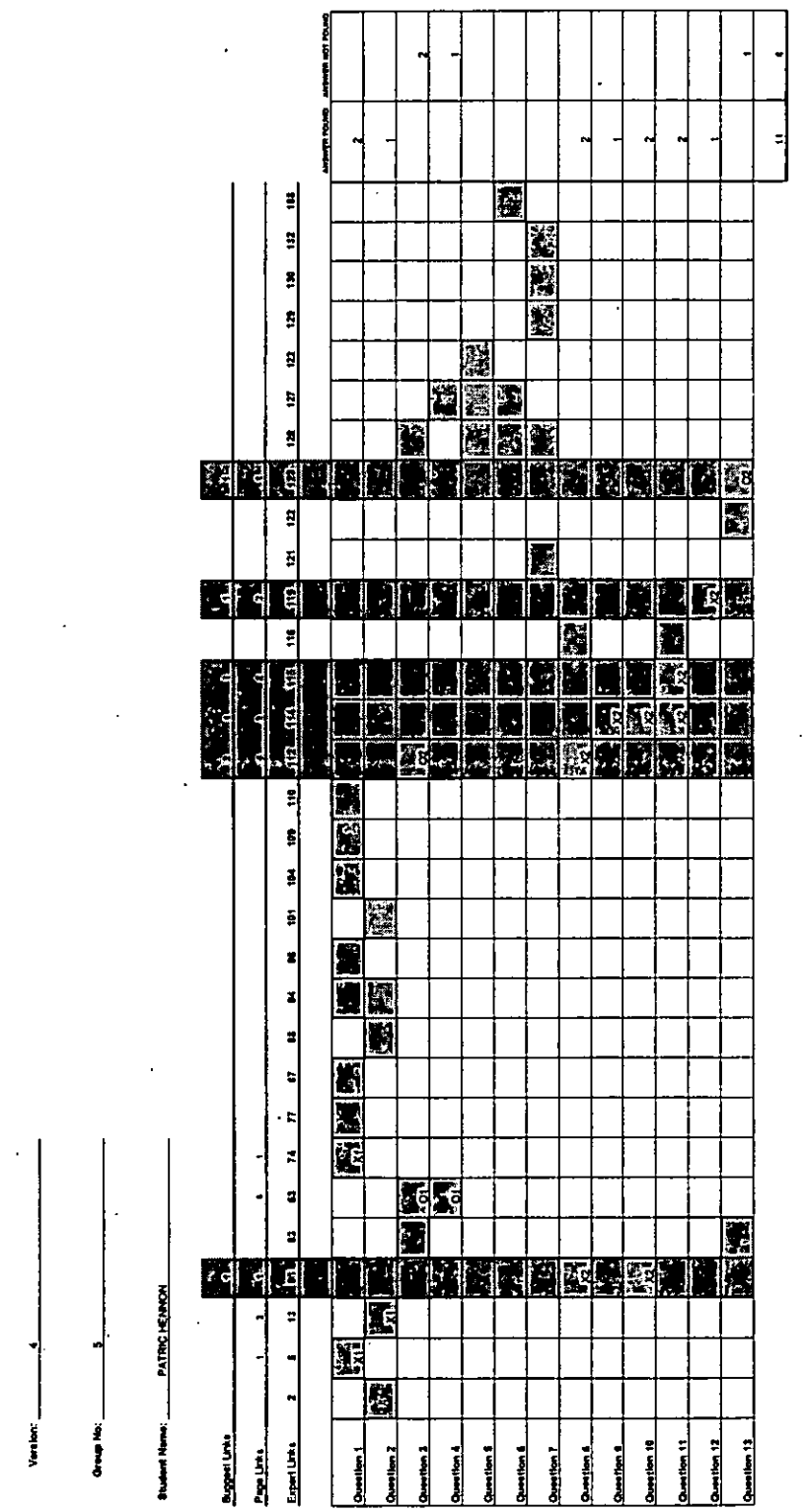
Suggest Links

Page Links 3 3 1 1 1 1 1 1 1 1 1

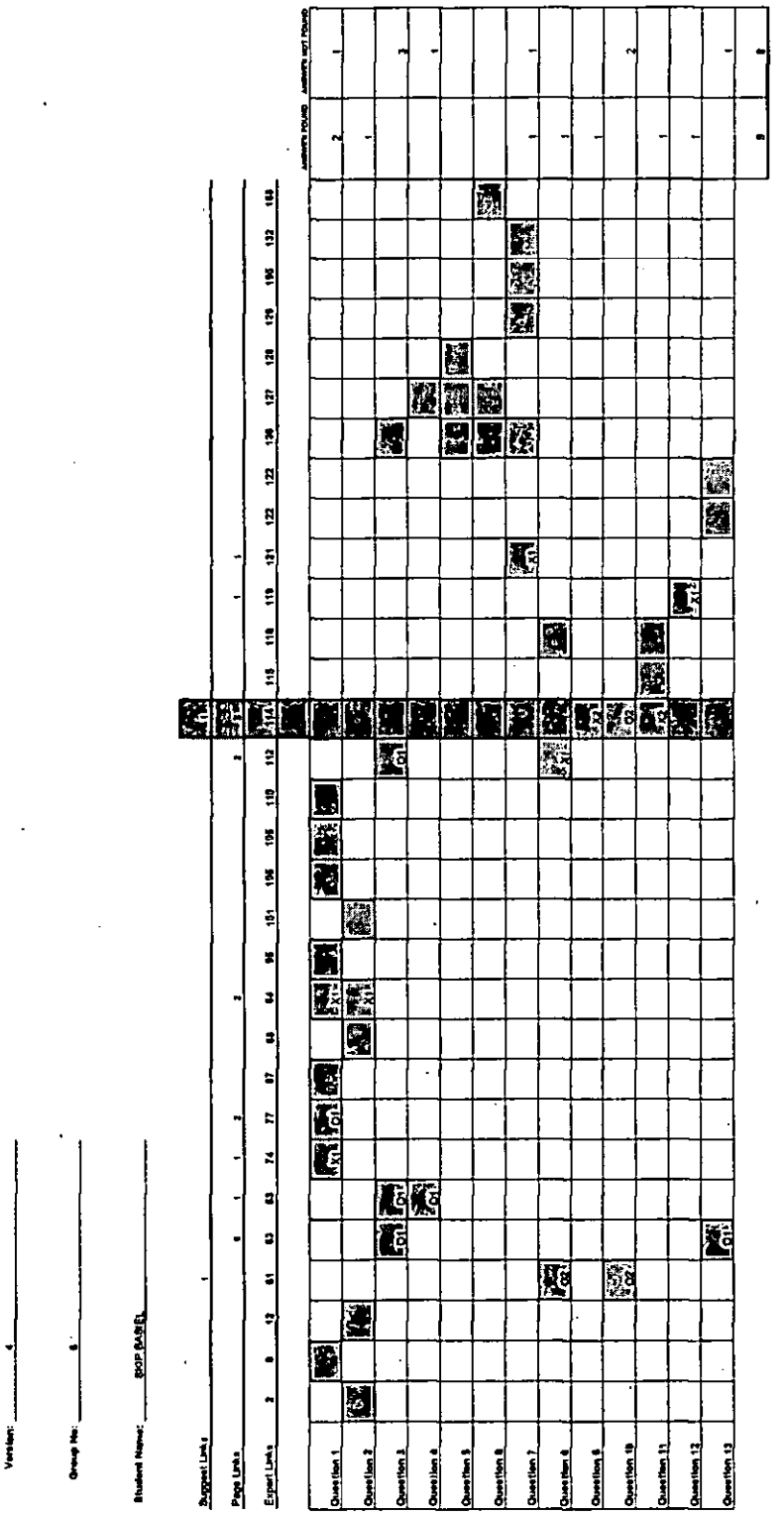
Expert Links 2 5 12 51 52 63 74 77 87 88 94 96 101 104 109 110 112 114 115 116 119 121 123 123 128 127 128 129 130 132 133

																																					ANSWER FOUND	ANSWER NOT FOUND
Question 1								X1				X1																										2	
Question 2																																							
Question 3																		X1									X1											2	
Question 4																																							
Question 5																											X1											1	
Question 6																											X1											1	1
Question 7																											X1											2	1
Question 8																		X1																				1	
Question 9																			X1																			1	
Question 10																			X1																			1	
Question 11																			X1	X1																		2	
Question 12																						X1																1	
Question 13																								X1	X1													1	
																																					15	2

Appendix D


Version: 4
Group No: 5
Student Name: PATRIC HENNON
Bugger Links
Page Links
Expert Links
Question 1
Question 2
Question 3
Question 4
Question 5
Question 6
Question 7
Question 8
Question 9
Question 10
Question 11
Question 12
Question 13
ANSWERS FOUND
ANSWERS NOT FOUND

Appendix D


Version:
Group No:
Student Name:
BOP BASE
Suggest Links
Page Links
Expert Links
Question 1
Question 2
Question 3
Question 4
Question 5
Question 6
Question 7
Question 8
Question 9
Question 10
Question 11
Question 12
Question 13
ANSWERS FOUND
ANSWERS NOT FOUND

Appendix E

E.1 Problems with the conceptual clustering algorithm

The conceptual clustering algorithm used in MLTutor contains a significant weakness as commented on by Hutchinson (1994). This is based on the initial sort step and is demonstrated in the following example.

Suppose we have a population of six objects described by 9 attributes as shown below.

	Attr 1	Attr 2	Attr 3	Attr 4	Attr 5	Attr 6	Attr 7	Attr 8	Attr 9
P1	0	1	1	0	0	1	0	0	1
P2	1	1	1	1	0	1	1	0	1
P3	1	1	1	0	1	1	1	0	1
P4	1	1	1	0	0	1	0	0	1
P5	1	1	1	1	0	1	0	0	1
P6	0	0	0	0	1	0	0	0	1

The first step of the clustering process is to develop a list of ordered pairs of objects and measure the distance, based on dissimilarity, between each object in the pair.

1. d (P1, P2) = 3
2. d (P1, P3) = 3
3. d (P1, P4) = 1
4. d (P1, P5) = 2
5. d (P1, P6) = 4
6. d (P2, P3) = 2
7. d (P2, P4) = 2
8. d (P2, P5) = 1
9. d (P2, P6) = 7
10. d (P3, P4) = 2
11. d (P3, P5) = 1
12. d (P3, P6) = 5
13. d (P4, P5) = 1
14. d (P4, P6) = 5
15. d (P5, P6) = 6

First approach

The next stage of the clustering process sorts the pairs into ascending order based on distance.

1. d (P1, P4) = 1
2. d (P2, P5) = 1
3. d (P3, P5) = 1
4. d (P4, P5) = 1
5. d (P1, P5) = 2
6. d (P2, P3) = 2
7. d (P2, P4) = 2
8. d (P3, P4) = 2
9. d (P1, P2) = 3
10. d (P1, P3) = 3
11. d (P1, P6) = 4
12. d (P3, P6) = 5
13. d (P4, P6) = 5
14. d (P5, P6) = 6
15. d (P2, P6) = 7

Cluster formation following the algorithm proceeds as follows.

Step 1:

Cluster 1 is created with the first pair {P1, P4}.

Cluster 1 currently contains: (P1, P4)

Step 2:

None of the points of the next pair {P2, P5} is in Cluster 1 and Cluster 2 is created.

Cluster 2 currently contains: (P2, P5)

Step 3:

One of the points of the next pair {P3, P5} is in Cluster 2 so, P3 is added to the Cluster 2.

Cluster 2 currently contains: (P2, P5, P3)

Step 4:

The first point of the next pair {P4, P5} is in Cluster 1 and the second point is in Cluster 2. The algorithm decides whether the two clusters should be amalgamated.

P4 is nearly central if the distance between P4 and any other point in the cluster is less then 2/3 of the diameter of that cluster. The diameter of a cluster is the maximum distance between two points in the cluster. For example the diameter is max {d (P1, P4) = 1}, diameter (Cluster 1) =

1. In this case *d* (P1, P4) is not less the 2/3 of the diameter and so P4 is not nearly central. As a consequence, Cluster 1 and Cluster 2 remain unchanged.

Step 5:

The next pair {P1, P5} does not result in any changes to the clusters after repeating the calculation of step4 on Cluster 2.

Step 6:

The next pair {P2, P3} is in Cluster 2 already.

Step 7, 8, 9, 10:

Repeats the calculation of step 4 and 5 and does not result in changes to the clusters.

Step 11:

One point of the next pair {P1, P6} is in Cluster 1 and P6 is added to the same cluster.

Cluster 1 currently contains: (P1, P4, P6)

Step 12:

Does not result in any changes to the clusters.

Step 13:

Both points of the next pair {P4, P6} are in Cluster 1.

Cluster 1 currently contains: (P1, P4, P6)

Step 14 and 15:

Do not result in any changes to the clusters.

The two clusters created during clustering are as follows.

Cluster 1: (P1, P4, P6)

Cluster 2: (P2, P5, P3)

Second approach

Suppose the sort step puts the pairs into the following order which reverses the position of the {P4, P5} and {P1, P4}.

1. d (P4, P5) = 1
2. d (P2, P5) = 1
3. d (P3, P5) = 1
4. d (P1, P4) = 1
5. d (P1, P5) = 2
6. d (P2, P3) = 2
7. d (P2, P4) = 2
8. d (P3, P4) = 2
9. d (P1, P2) = 3
10. d (P1, P3) = 3
11. d (P1, P6) = 4
12. d (P3, P6) = 5
13. d (P4, P6) = 5
14. d (P5, P6) = 6
15. d (P2, P6) = 7

Cluster formation following the conceptual clustering algorithm proceeds as follows.

Step 1:

Cluster 1 is created with the first pair {P4, P5}.

Cluster 1 currently contains: (P4, P5)

Step 2:

One of the points of the next pair {P1, P4} is in Cluster 1 and P1 is added to the same cluster.

Cluster 1 currently contains: (P4, P5, P1)

Step 3

One of the points of the next pair {P2, P5} is in Cluster 1 and P2 is added to the same cluster.

Cluster 1 currently contains: (P4, P5, P1, P2)

Step 4

One of the points of the next pair {P3, P5} is in Cluster 1 and P3 is added to the same cluster.

Cluster 1 currently contains: (P4, P5, P1, P2, P3)

Step 5

Both points of the next pair {P1, P5} are in Cluster 1 and Cluster 1 remains the same.

Cluster 1 currently contains: (P4, P5, P1, P2, P3)

Step 6, 7, 8, 9 and 10

Repeats the calculation of step 5 and does not result in any changes to Cluster 1.

Step 11

One of the points of the next pair {P1, P6} is in Cluster 1 and P6 is added to the same cluster.

Cluster 1 currently contains: (P4, P5, P1, P2, P3, P6)

Step 12, 13, 14 and 15

Repeats the calculation of step 5 and does not result in any changes to Cluster 1.

One cluster is created during clustering process as follows.

Cluster 1: (P4, P5, P1, P2, P3, P6)

The conceptual clustering algorithm is thus sensitive to the sort step as the order of the equally distant pairs can cause different clusters to be generated.

E.2 Pre-clustering Step 1

```
clust07page026000000000000000000000000000001000001000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx *
clust07page024000000000000000000000000000100001000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx *
clust07page029000000000000000000000000000000000001000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx *

clust13page050000000000000000000000000000000000000000000000000010000000000000000000000000000000001000000000000000000000000000000010000000000000000000000000000000000000000000000000 xxx *
clust13page046000000000000000000000000000000000000000000000000100000000000000000000000000000000000100000000000000000000000000000000000000000000000000000000000000010000000000000000 xxx *

clust17page062000000000000000000000001000000000000000000000000000000000000000000000000000000000010010000000000000000000000010000000000000000000100010000000000000000000000 xxx *
clust17page063000000000000000000000000000000000000000000000000000000001000000000000000000000000000001001000000000000000000000001000100000000000001000100000000000000000000 xxx *

clust21page128000000001110000000100000000100000000000000001001000000000000000000100000000001000000000000010000010000010010000000000000000000000001000000000000000000000000000 xxx *
clust21page125000000000001100000000000000001000000000000000000001000000000000000000000000000000000000000000001000000000000000000100000000000000000000000000100000000000000000 xxx *
clust21page124000000000001000000000000000001000000000000000000001000000000000000000000000000000000000000000001110000000000000000000000000000000000000000000000000000000000000 xxx *
clust21page129000000000011000000010000000010000000000000001000000000000100000010000000000100000010001000010000000010000000000000010000000000000000101000000000 xxx *
clust21page126000000000001100000000000000001000000000000000000001000000000000000000000000000000000000001000001110001001000000000000000000000000000000000000000000000000000000 xxx *

clust27page002100000000100000000100000001000000000000000000000000000000000001001000000000000000000000000000000000000000000000000000001000001000000000000000000000000000 xxx *
clust27page012100000000100000000000000000000000000000000001000000000000000000000000010000000000000000000000000000000000000000000000000000010000010000000000000000000001000000 xxx *

clust28page088101010000000000000000000000000000000000100010000000000000000000000110000000000000000000000001001001000000000000000000001000000000000000000000000000000 xxx *
clust28page101100010000000000000000000000000000000100000000000000000000000100100000000000000000000000000000000000000000000000010100000000000000000000100000000 xxx *
clust28page109101010000000000000000000000000000000001000000000000000000000000001001000000000000000000000000000000100000000000000000000101000000000000000000000000000000 xxx *
clust28page104101010000000100000000000000000000000000001000000000000001000000100000000000000000000000000000001001000000000000000000000001000000000000000000000000000 xxx *
clust28page094111011000000000000000000000000000000100000000100000000000000110000010110000000001000000000000000101000100000000000100000000110010000000000010000001000000 xxx *

clust29page131000000000101000000000100000000100000000000110000010000000000000000000000000000000000010000001000000000000000010000001000000000000000000000000000000000000 xxx *
clust29page130000000000101000000000000000000100000000000100000100000010000000000000000000000000000000000000000000000000100000000000000000001000000000000000000000000000000000000 xxx *
clust29page132000000000111100000000010000000010000000000100000000000000000000000000000000000000000000000100001000000000000000000000000000000000000000000000000000000000000000000 xxx *

clust30page003100000000100000000000000000000000000000000000000000000010000000001000000000000000000000000000000000000000000000000000000100000000000000000000000000000000010000 xxx *
clust30page010100000000100001000101000000000000000000000000100000000000000001000000000100000000000000000000000000000100000000000000000011001000000000000000000010000 xxx *
clust30page006100000000100000000000000000000000000000000000001000000000000010000000001000000000000000000000000000000000000000000000000000001000000000000000000000000000 xxx *
clust30page004100000000100000000000000000000000000000000000000010001000000000000000000000000000000000000000000000000001000000000000000001000000000000000000000000010000 xxx *

clust32page031000000000100000000000000000000100000000000000000000000100000000000000000000000000000000000000000000000000000000000001100000000000000000000000000000000000000 xxx *
clust32page071000000000100000000001000000000000000000000000000000000001000000000000000100000000000000000000000100000000000000000010000000000000000000000000000000000000000 xxx *
clust32page016100000000100000000000000000000000000000000000000000000000000000000000000000000000000001000000001000000000000000000100000000000100000000000000000000000000 xxx *
clust32page100100000000100001000100100000000000000000010000000100000000000000000000010000000000000000000000000000100000000000100010000000000010000000001000000000000000 xxx *
```

from this point re-clustering applied

```
clust38page042000000000000000000000000000100001000000000000000000000000000000000000000000000000000011000000000000000000000000000000000000000000000000000000001000000000000000000000000 xxx *
clust38page043000000000000000000000000000010000000000000000000000000000000000000000000000000000000011000000000000000000000001000000000000000000000000000000000000000000000001000000 xxx *
clust38page044000000000000000000000000000100001000000000000000000000000000000000000000000000001000000110000000000000000000000000000000000000000000000000000000000000000000000000000 xxx *
clust38page045000000000000000000000000001000001000000000000000000000000000000000000000000000001000000110000000000000000000000000000000000000000000000010000000000010000000000001000000 xxx *
clust38page047100000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000010000000000100000000000000100000000000100000011000000 xxx *
clust38page048100000000000000000000000000000000000000000000000001000000100000000000000000000000000110000000000000000000000000000000000000000000100001000100000000000000001000000 xxx *
clust38page049000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000 xxx *
clust38page051000000000000000000000000000000000000000000000000100000001000000000000000100000010000000000000000000000000000000000000000000000000001000000000000000000000000000 xxx *
clust38page052000000000000000000000000000000000000000000000000000000000000000000000000000000000010000000000000000000000000000000000000000000000100000000000000000000000000 xxx *
clust38page053000000000000000000000000001000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx *
clust38page054000000000000000000000000000000000000000000000001000000010000000000000000000001011000000000000000000000000000010000000000000100000010001000000000000000100000 xxx *
clust38page055000000000000000000000000000000000000000000000000000000000000000000000000000000000110000000000000000000000000000000000000000010000000000000000000000000000000000000 xxx *
clust38page056000000000000000000000000000000000000000000000000000000000000000000000000000000001011000000000000000000000000000000000000000000000000000000000000000000000000000 xxx *
clust38page057000000000000000010010000000000000000000000000000000000000000000000000000000100010000000000000000001001000000000000000000000000000000000000000000000100000000 xxx *
clust38page058000000000000000000000000000000000000000000000000000000000000000000000010000000000000000000000000000000000000000000000000000000000000000000000000100000000 xxx *
clust38page059000000000000000000000000101000100000000000000000000000000000000000000000000000000000000000000000000000000000000001000001000000000000100010000000000000000000 xxx *
clust38page060000000000000000000000001001000100000000000000000000000000000000000000000000000010001000000000000000000000000100000000100000000000000000000100000000000 xxx *
clust38page061000000000000000000000001000001000000000000000000000000000000000000000000000000000011000000000000000000000001000000000000000000000100000000000000001000 xxx *
clust38page064000000000000000000000000000001000000000000000000000000000000000000000000000000000001001000000000000000000000000000000000000000000000000000000000000000000000000 xxx *
```

clust38page037
clust38page036
clust38page035
clust38page034
clust38page033
clust38page032
clust38page030
clust38page028
clust38page027
clust38page025
clust38page023
clust38page022
clust38page021
clust38page020
clust38page019
clust38page017
clust38page016
clust38page015
clust38page014
clust38page013
clust38page011
clust38page009
clust38page008
clust38page007
clust38page005
clust38page001
xxx

E.3 Pre-clustering Step 2

```
clust11page083000000000000000001000000000000000000000000000010000000000000000000000000000000000010000000000000001000000000001000000001000010000000 xxx •
clust11page082000000000000001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust11page081000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust11page084000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust11page058000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust11page078000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust11page077000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust11page076000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust11page075000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust11page074000000000000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust11page017000000010000000000000000000000000101000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust11page057000000000000000100100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust11page085000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust11page086000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust11page070000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust11page087000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust11page047100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •

clust12page068100000001000010100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust12page066000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust12page069000000001000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust12page120000000000000010000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust12page060000000000000000000100100010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust12page065000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •

from this point re-clustering applied
clust23page092000000001000000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page093000000001000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page095001010000000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page096000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page097000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page098100010000000000000000000000000000001001000000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page099100001000000000000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page102000000001000000000010000000000000000010000000000000000010100000001000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page103100010000000011000001000000000000000010000000000000000000000000110000000000000000000000001001000010000000000001000000000000000000000 xxx •
clust23page105101100000000010010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page106001101000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page107100100000000000010000000000000000010010000000000000000010010000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page108000000000000000000000000000000000010010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page110000100000000100000000000000000000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page111000000000000000000001000000000000100000000000000000000000000100000000000000100110010000000000010000000000000000000000000000000000 xxx •
clust23page112000000000000010000001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page113000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page114000000000000010000001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page115000000100000000000011000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page116000000000000000000010000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page117000000100000010001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page118000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page119000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page121000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page122000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page123000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page127000000000110000001000000000100000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page133000000000110000000000000001100000000000001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page041000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust23page040000000001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
```

E. 4 Pre-clustering Step 3

```
clust06page038000000000000000000000000000000000000010000000000000000000010000000000000000000000000000010000000000000000000000000000000000000000000000 xxx
clust06page039000000000000000000000000000000000000000000000000000000000010000000000000000000000000000010000000000000000000000000000000000000000000000 xxx
clust06page040000000001000000000000000000000000000010000000001000000000010000000000000000000000000010000000001000000000000000000000000000000000000 xxx
clust06page037000000000000000000000000000000000110000000000000000000010000000000000000000000000000010000000000000000000000000000000000000000000000 xxx
clust06page092000000001000000100000000000000000000000000000000000000000010000001001000000000000010000000000000000000100000000000000000000000000 xxx

clust09page121000000000000010000000000000000000000000000000000000000000000000000000000000000000000000111000000000000000000010000000000000000000000000 xxx
clust09page113000000000000010000000000000000000000000000000000000000000000000000000000000000000000001101000000000000000000000000000000000000000000000 xxx


from this point re-clustering applied
clust19page102000000001000000000010000000000000000000010000000000000000010100000000100000000000000000000000000000000000000000000000000000000000000 xxx
clust19page103100010000000110000010000000000000000000100000000000000000000000000000011000000000000000000000000000000000000000000000000000000000000 xxx
clust19page105101100000000100100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page106001101000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page107100100000000000010000000000000000000010010000000000000000010010000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page108000000000000000000000000000000010010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page110000100000001100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page111000000000000000000001000000000001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page112000000000000010000001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page114000000000000100000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page115000000100000000000011000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page116000000000000000000001000010000000000000000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page117000000100000010001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page118000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page119000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page122000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page123000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page127000000000110000001000000001000000000100000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page133000000000110000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page041000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page036100000000000000001001000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page035000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page034000000000000000000001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page033000000001000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page032100000010000000001001000001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page030000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page028000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page025000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page023000000000000000000000001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page022000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page021000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page020000000000000000000000001000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page019000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page016000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page015000000000000000000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page014000000010000000000000000000000000010000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page013101001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page011100000001000000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page009101000001000000001000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page008100000001000000110000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page007101000001000000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page005001100001000100100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page001100000001000000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page091100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page090100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
clust19page089110000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx
```

```
clust19page0801000000001000000000000000000000000010000100000000000000000001000000000000000000000000000000000000000000000000000000000000000000000000000000000110000000000 xxx •
clust19page0791000000001000000000000000000000000010000100000000000000000001000000000000000000000000000000000000000000000000000000000000000000000000000000000110000000000 xxx •
clust19page0730000000010000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000010000010000001001000001011000000 xxx •
clust19page0720000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000100000000000000000000000000000010000000110000 xxx •
clust19page0670000000000000000000001000010000000000000000000000000000000000000000000000000000000000011000000000000000000000001000000000100000000000000000000000000000 xxx •
clust19page0640000000000000000000000000000001000000000000000000000000000000000000000000000000000000010010000000000000000000000000000000000000000000000000000000000000 xxx •
clust19page0610000000000000000000000010000010000000000000000000000000000000000000000000000000000000011000000000000000000000000100000000000000000000000010000000000000001000 xxx •
clust19page0590000000000000000000000010100010000000000000000000000000000000000000000000000000000000000000000000000000000000001000000100000000000010001000000000000000000 xxx •
clust19page0560000000000000000000000000000000000000000000000000000000000000000000000000000000010110000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust19page0550000000000000000000000000000000000000000000000000000000000000000000000000000000000110000000000000000000000000000000000000000001000000000000000000000000000 xxx •
clust19page0540000000000000000000000000000000000000000000000001000000010000000000000000000000010110000000000000000000000000000001000000000000010000001000100000000000000100000 xxx •
clust19page0530000000000000000000000000100000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust19page0520000000000000000000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000001000000000000000000 xxx •
clust19page0510000000000000000000000000000000000000000000001000000010000000000000000000000100000001100000000000000000000000000000000000000000000000000100000000000000000000 xxx •
clust19page0490000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000010000000000000000000 xxx •
clust19page0481000000000000000000000000000000000000000000001000000010000000000000000000000000000001100000000000000000000000000000000000000000100001000100000000000010000000 xxx •
clust19page0450000000000000000000000000010000001000000000000000000000000000000000000000000100000001100000000000000000000000000000000000000000010000000000010000000000001000000 xxx •
clust19page0440000000000000000000000000001000010000000000000000000000000000000000000000000100000001100000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust19page0430000000000000000000000000001000000000000000000000000000000000000000000000000000000001100000000000000000000000100000000000000000000000000000000000000000001000000 xxx •
clust19page0420000000000000000000000000001000010000000000000000000000000000000000000000000000000001100000000000000000000000000000000000000000000000000100000000000000000000 xxx •
clust19page0991000010000000000000000000000000000001000000000000000000000000000000100010100000000000000000000000000000001000000000000000000000000000000000000000000000000 xxx •
clust19page0981000100000000000000000000000000000000010010000000000000000000000010000000000000000000000000000000000000001000000000100000000000010000000000000000000000000000 xxx •
clust19page0970000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000100000000000001000000000000000000000000000 xxx •
clust19page0960000000000000000000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000100100000000000000000001000000000000000000000000000 xxx •
clust19page0950010100000000000000000000000000000000001000000000000000000000000000000000000000000000100000000000000000000000000010000000000000000000000000000000000000100000000000 xxx •
clust19page0930000000001000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000101000000000000001000000000000 xxx •
clust19page0270000000000000000000000000000000000000000000000000000000000000000000000000000000010110000000000000000000000000000000000000000000000000000000000000000000 xxx ••
```

E. 5 Pre-clusteringStep 4

```
clust02page0210000000000000000000000000000000000000000000000000000000000000000001001000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust02page0270000000000000000000000000000000000000000000000000000000000000000001011000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust02page0550000000000000000000000000000000000000000000000000000000000000000000011000000000000000000000000000000001000000000000000000000000000000000000000 xxx •
clust02page0560000000000000000000000000000000000000000000000000000000000000000001011000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust02page0300000000000000000000000000000000000000000000000000000000000000000001011000000000000000000000000000000000000000000000000000000000000000000000000 xxx •

from this point re-clustering applied
clust16page1080000000000000000000000000000000000001001000000000000000000000000000000000000000000000000000000000010000000000001010000000000000000000000000 xxx •
clust16page1100001000000001000000000000000000000000001000000000000000000000000000000000100000000000000100000000000000000000000000000000000000000000000 xxx •
clust16page1110000000000000000000010000000000000010000000000000000000000000000010000000000001001100100000000000001000000000000000000000000000001000001000 xxx •
clust16page1120000000000000001000000100000000000000000000000000000000000000000000000000000001000100000001000000000001001001000000000000000000000000000000000 xxx •
clust16page1140000000000000001000000100000000000000000000000000000000000000000000000000000011000000000000000000000001000000001000000100000000000001000001000 xxx •
clust16page1150000001000000000000011000000000000000000000000000000000000000000000000000000011000000000000000000010100000000000000000000000000000000000000 xxx •
clust16page1160000000000000000000010000100000000000000000000000100000000000000000000000000010011000000000000000000001000000001000000000000000000000000000000 xxx •
clust16page1170000001000000100010000000000000000000000000001000000100000000000000000000000000001000000000000000000000000000001000000000000000000001001000000 xxx •
clust16page1180000000000000010000000000000000000000000000000000000000000010000000000000000100000000000000000000000010000000010000000000000000000000000000 xxx •
clust16page1190000000000000010000000000000000000000000000000000000000000000000000000001000000000110000000000000000000000000000000000000000000000000001000 xxx •
clust16page1220000000000000010000000000000000000000000000000000000000000000000000000000000100000000000000000000000010000000000000000000000000000000001000 xxx •
clust16page1230000000000000000000000000000000000000000010000000000000000000000000000000000100000000000000000000000010000000000000000000000000000000000000 xxx •
clust16page1270000000001100000010000000001000000000100000100000000000000000000001000000000001000101000010000000000000000001000000100000000000000000001000000000 xxx •
clust16page1330000000001100000000000000001100000000000000100000000000000000000000001000001000001000000000000000000000000000000000000000000000000000000000 xxx •
clust16page0410000000100000000000000000000000000000000000000000000000000000000000110000000000000000000001000000000000000000000000000000001001000000001000000 xxx •
clust16page0361000000000000000001001000000000000000000001000000000000000000000000000000000000000000000000000000000000000000000000000000001110000000000000000000 xxx •
clust16page0350000000000000000000000000000000000000100000000000000000000000000000000000000000000000000000000000010000000000000000001010000000000000000000000 xxx •
clust16page0340000000000000000000001000000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000000001100000000000000000000 xxx •
clust16page0330000000001000100000000000000000000000000000000010000000010000000000000000000000000000000000000000000100000000000000000000111000000000000000000 xxx •
clust16page0321000000100000000010010000010000000000000001000100000000100000010000010000000001000000000000000000000011000010001000000001010000000000001100000 xxx •
clust16page0280000000000000000000000000000000000000000000000000000000000000000000000000000010000000000000000000000000000000000000000000000000000000000000 xxx •
clust16page0250000000000000000000000001000000000000000000000000000000000000000000000000001011000000000000000000000000000000000000000000000000000000000000 xxx •
clust16page0230000000000000000000000001000000000000000000000000000000000000000000000000001011000000000000000000000000000000000000000000000000000000000000 xxx •
clust16page0220000000000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000000 xxx •
clust16page0200000000000000000000000001000010000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000000 xxx •
clust16page0190000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000000000000000100000000000000000100000000000 xxx •
clust16page0160000000000000001000000000000000000100000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000 xxx •
clust16page0150000000000000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000 xxx •
clust16page0140000000100000000000000000000000000000100001000000000000000000000000000000000000000000000000000000000100000000000000000000000000000001100000 xxx •
clust16page0111000000010000001000000000000000000000000000000010000000100000000000000000000000000000000000010000000000000000001000000000000000000100000000000 xxx •
clust16page0091010000010000000010000000100000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000 xxx •
clust16page0081000000010000001100000000000000000000000000000001000000010000000000010000000000000000000100000000000000000000000000000000000000000001000000000 xxx •
clust16page0071010000010000001000000000000000000000000000000010100000001000000000000000000000000000000010000000000000000000000000000000000001100000000000 xxx •
clust16page0050011000010001001000000000000000000000000000000000100000001000000000000000000000000000000110000000000000000000000000000000001100000000000 xxx •
clust16page0011000000010000001000000000000000000000000000000001000000010000000000000000000000000000000100000000000000000000000000000001000000000000 xxx •
clust16page0911000000000000000000000000000000000100000000000000000000010001100000000000000011000000000000000000000000000000000000000000000000000000000 xxx •
clust16page0901000000000000000000000000000000001000000000000000000000000100011000000000000000110000000000000000000000000000000000000000000000000000000 xxx •
clust16page0891100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx •
clust16page0801000000010000000000000000000000000100001000000000000001000000000000000000000000000000000000000000000000000000000000000000000000011000000 xxx •
clust16page0791000000010000000000000000000000000000010000100000000000000100000000000000000000000000000000000000000000000000000000000000000000011000000 xxx •
clust16page0730000000100000000000000000000000000000000000000000000000000000000000000000010000000000000000000000000000000000000100000100000100100000101100000 xxx •
clust16page0720000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000010000000110000 xxx •
clust16page0670000000000000000000000100001000000000000000000000000000000000000000000000011000000000000000000000010000000000100000000000000000000000000000 xxx •
clust16page0640000000000000000000000000001000000000000000000000000000000000000000000000000001001000000000000000000000000000000000000000000000000000000000 xxx •
clust16page0610000000000000000000001000001000000000000000000000000000000000000000000000000011000000000000000000000100000000000000000100000000000000001000 xxx •
clust16page0590000000000000000000001010001000000000000000000000000000000000000000000000000000000000000000000000100000100000000000010001000000000000000000 xxx •
clust16page0540000000000000000000000000000000000000010000000100000000000000000000101100000000000000000000000000000100000000000010000001001000000000000100000 xxx •
clust16page0530000000000000000000000001000000000000000000000000000000000000000000000000010000000000000000000000000000000000000000000000000000000000000000 xxx •
clust16page0520000000000000000000000000000000000000000000000000000000000000000000000000000010000000000000000000000000000000000000000000000000100000000000000 xxx •
clust16page0510000000000000000000000000000000000000010000000100000000000000100000011000000000000000000000000000000000000000000000100000000000000000000000 xxx •
```

```
clust16page0490000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000 xxx *
clust16page0481000000000000000000000000000000000000000000000100000010000000000000000000000000001100000000000000000000000000000000000000000000100001000100000000000001000000 xxx *
clust16page0450000000000000000000000000100000100000000000000000000000000000000000010000001100000000000000000000000000000000000000010000000000001000000000000001000000 xxx *
clust16page0440000000000000000000000000100001000000000000000000000000000000000000010000001100000000000000000000000000000000000000000000000000000000000000000000000 xxx *
clust16page0430000000000000000000000000100000000000000000000000000000000000000000000000000011000000000000000000000100000000000000000000000000000000000000001000000 xxx *
clust16page0420000000000000000000000000100001000000000000000000000000000000000000000000000011000000000000000000000000000000000000000000010000000000000000000000000 xxx *
clust16page0991000010000000000000000000000010000000000000000000000000000000010001010000000000000000000000000001000000000000000000000000000000000000000000000000000 xxx *
clust16page0981000100000000000000000000000000000010010000000000000000000010000000000000000000000000000000000001000000010000000000001000000000000000000000000000000 xxx *
clust16page0970000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000010000000000000100000000000000000000000000 xxx *
clust16page0960000000000000000000000000000000000000000000000000000000000000000010000000000000000000000000000010010000000000000000001000000000000000000000000000 xxx *
clust16page0950010100000000000000000000000000100000000000000000000000000000000000000000000000100000000000000000000000100000000000000000000000000000000001000000000000 xxx *
clust16page0930000000001000000000000000000000001000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000010100000000000001000000000000 xxx *
clust16page0131010010000000000000000000000000000000000000000000001000000000000000100000000000000000000000100000000000000000100001000000000000000000000000000 xxx *
clust16page1071001000000000000100000000000000000010010000000000000000000010010000000000000000000000000000001000000000010000000000000000000000000000000001000000 xxx *
clust16page1060011010000000000010000000000000000000000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000001000000 xxx *
clust16page1051011000000001001000000000000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000100000000000000000000000000 xxx *
clust16page1031000100000001100000100000000000000000100000000000000000000000000000110000000000000000000000000100100001000000000000000100000000000000000000000000 xxx *
clust16page1020000000001000000000100000000000000000100000000000000000101000000001000000000000000000000000000000000000000000000000100010000000000000100000000 xxx **
```

E. 6 Pre-clustering Step 5

```
clust08page015000000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000 xxx *
clust08page014000000001000000000000000000000000001000001000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000001100000 xxx *
clust08page016000000000000000100000000000000000001000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000 xxx *

from this point re-clustering applied
clust16page112000000000000001000000100000000000000000000000000000000000000000000000000001000100000001000000000001001001000000000000000000000000000000000000 xxx *
clust16page114000000000000001000000100000000000000000000000000000000000000000000000000011000000000000000000001000000001000001000000000000001000001000 xxx *
clust16page115000000100000000000000110000000000000000000000000000000000000000000000000011000000000000000000101000000000000000000000000000000000000000 xxx *
clust16page116000000000000000000000100001000000000000000000000000001000000000000000000000001001100000000000000000000001000000001000000000000000000000000000000 xxx *
clust16page117000000100000010001000000000000000000000000001000000100000000000000000000000001000000000000000000000000000001000000000000000000001001000000 xxx *
clust16page118000000000000001000000000000000000000000000000000000000000000100000000000000001000000000000000000000001000000001000000000000000000000000000000 xxx *
clust16page119000000000000001000000000000000000000000000000000000000000000100000000000011000000000000000000000000000000000000000000000000000000001000 xxx *
clust16page122000000000000001000000000000000000000000000000000000000000000000000000000001000000000000000000000001000000000000000000000000000000000001000 xxx *
clust16page123000000000000000000000000000000000000000000001000000000000000000000000000000001000000000000000000000001000000000000000000000000000000000000000 xxx *
clust16page127000000000011000000100000000010000000001000001000000000000000000000100000000000100010100001000000000000000000001000000100000000000000001000000000 xxx *
clust16page133000000000011000000000000000001100000000000000100000000000000000000000000000100000100000100000000000000000000000000000000000000000000000000000 xxx *
clust16page041000000001000000000000000000000000000000000000000000000000000000000000000001100000000000000000001000000000000000000000000000000001001000000001000000 xxx *
clust16page036100000000000000001001000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000000000001110000000000000000000 xxx *
clust16page035000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000100000000000000000001010000000000000000000 xxx *
clust16page034000000000000000000000100000000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000000001100000000000000000000 xxx *
clust16page033000000001000010000000000000000000000000000001000000001000000000000000000000000000000000000000000000100000000000000000001110000000000000000000 xxx *
clust16page032100000010000000001001000001000000000000001000100000000100001000001000000000100000000000000000000011000001000100000000101000000000000011000000 xxx *
clust16page028000000000000000000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000 xxx *
clust16page025000000000000000000000001000000000000000000000000000000000000000000000001011000000000000000000000000000000000000000000000000000000000000000000 xxx *
clust16page023000000000000000000000001000000000000000000000000000000000000000000000001011000000000000000000000000000000000000000000000000000000000000000000 xxx *
clust16page022000000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000000000000 xxx *
clust16page020000000000000000000000001000010000000000000000000000000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000 xxx *
clust16page019000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000100000000000000100000000000 xxx *
clust16page011100000000100000010000000000000000000000000000000000000100000000000000000000000000000000000000000010000000000000000001000000000000000001000000000000 xxx *
clust16page009101000001000000001000000100000000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000 xxx *
clust16page008100000001000000110000000000000000000000000000000001000000100000000000100000000000000000001000000000000000000000000000000000000100000000000 xxx *
clust16page007101000001000000100000000000000000000000000000000000101000000010000000000000000000000000000000100000000000000000000000000000000001100000000000 xxx *
clust16page005001100001000100100000000000000000000000000000000010000000100000000000000000000000000000011000000000000000000000000000000000001100000000000 xxx *
clust16page001100000001000000100000000000000000000000000000000010000000100000000000000000000000000000010000000000000000000000000000000000001000000000000 xxx *
clust16page091100000000000000000000000000000000100000000000000000000100011000000000000000110000000000000000000000000000000000000000000000000000000000000 xxx *
clust16page090100000000000000000000000000000000100000000000000000000100011000000000000000110000000000000000000000000000000000000000000000000000000000000 xxx *
clust16page089110000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 xxx *
clust16page080100000001000000000000000000000001000010000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000110000000 xxx *
clust16page079100000001000000000000000000000001000010000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000110000000 xxx *
clust16page073000000010000000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000001000001000000100100000101100000 xxx *
clust16page072000000000000000000000000000000000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000100000001100000 xxx *
clust16page067000000000000000000000100001000000000000000000000000000000000000000000000000110000000000000000000010000000010000000000000000000000000000000 xxx *
clust16page064000000000000000000000000001000000000000000000000000000000000000000000001001000000000000000000000000000000000000000000000000000000000000000000 xxx *
clust16page061000000000000000000000001000001000000000000000000000000000000000000000000011000000000000000000000001000000000000000000100000000000001000 xxx *
clust16page059000000000000000000000101000100000000000000000000000000000000000000000000000000000000000000000000100000100000000000001000100000000000000000000 xxx *
clust16page054000000000000000000000000000000000010000000100000000000000000000101100000000000000000000000000100000000000100000010001000000000000000100000 xxx *
clust16page053000000000000000000000001000000000000000000000000000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000 xxx *
clust16page052000000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000001000000000000000000 xxx *
clust16page051000000000000000000000000000000000000000010000000100000000000000001000001100000000000000000000000000000000000000001000000000000000000000000000 xxx *
clust16page049000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000100000000000000000000000 xxx *
clust16page048100000000000000000000000000000000000000100000010000000000000000000000011000000000000000000000000000000000000000001000010001000000000000001000000 xxx *
clust16page045000000000000000000000001000001000000000000000000000000000000000001000000110000000000000000000000000000000000000001000000000010000000000001000000 xxx *
clust16page044000000000000000000000000100001000000000000000000000000000000000001000000110000000000000000000000000000000000000000000000000000000000000000000 xxx *
clust16page043000000000000000000000001000000000000000000000000000000000000000000011000000000000000000001000000000000000000000000000000000000000001000000 xxx *
clust16page042000000000000000000000000100001000000000000000000000000000000000000000000110000000000000000000000000000000000000000100000000000000000000000000 xxx *
clust16page099100001000000000000000000000000001000000000000000000000000000000100010100000000000000000000000000001000000000000000000000000000000000000000000 xxx *
clust16page098100010000000000000000000000000001001000000000000000000001000000000000000000000000000000000000100000001000000000001000000000000000000000000000 xxx *
clust16page097000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001000000000000010000000000000000000000000 xxx *
```

```
clust16page09600000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000001001000000000000000000001000000000000000000000000000 xxx *
clust16page09500101000000000000000000000000001000000000000000000000000000000000000001000000000000000000000001000000000000000000000000000000001000000000000 xxx *
clust16page09300000000100000000000000000000001000000000000000000000000000000000000000000000000000000000010000000000000000000010100000000000001000000000000 xxx *
clust16page01310100100000000000000000000000000000000000000000000100000000000000001000000000000000000000001000000000000000000100001000000000000000000000000000 xxx *
clust16page10710010000000000000100000000000000000010010000000000000000001001000000000000000000000000000000001000000000100000000000000000000000000000001000000 xxx *
clust16page10600110100000000000100000000000000000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000000000001000000 xxx *
clust16page10510110000000001001000000000000000000000000000000000000000000000000000000000000000000000000001000000000000000000001000000000000000000000000000 xxx *
clust16page10310001000000001100000100000000000000000010000000000000000000000000000011000000000000000000000000001001000010000000000000001000000000000000000000000 xxx *
clust16page10200000000100000000001000000000000000000100000000000000000000101000000000100000000000000000000000000000000000000000001000100000000000000100000000 xxx *
clust16page11100000000000000000000010000000000000001000000000000000000000000001000000000000010011001000000000000010000000000000000000000000000000001000001000 xxx *
clust16page11000010000000001000000000000000000000000001000000000000000000000000000000001000000000000010000000000000000000000000000000000000000000000000000 xxx *
clust16page10800000000000000000000000000000000000000001001000000000000000000000000000000000000000000000000000000000000000001000000000000101000000000000000000000000000 xxx **
```

E. 7 Pre-clustering Step 6

```
clust15page10510110000000001001000000000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000001000000000000000000000000000 xxx *
clust15page10310001000000011000001000000000000000001000000000000000000000000000000110000000000000000000000000000100100001000000000000000100000000000000000000000000000 xxx *
clust15page10200000000010000000000100000000000000000100000000000000000001010000000001000000000000000000000000000000000000000000001000100000000000000100000000 xxx *
clust15page11100000000000000000000000100000000000000100000000000000000000000000000100000000000100110010000000000000100000000000000000000000000000000000001000001000 xxx *
clust15page11000010000000001000000000000000000000000000000001000000000000000000000000000000001000000000000010000000000000000000000000000000000000000000000000000000 xxx *
clust15page10800000000000000000000000000000000000000100100000000000000000000000000000000000000000000000000000000000000000000000010000000000010100000000000000000000000000 xxx **
```

E. 8 Pre-clustered 133 pages

For the implementation of pre-clustering in versions 3 and 4 of MLTutor (see chapter 7), all 133 pages available within the system were clustered applying the conceptual clustering algorithm used in the dynamic variants of MLTutor. This resulted in 16 clusters being formed.

Cluster Number	Pages in Cluster
C1	clust01page026
	clust01page024
	clust01page029
C2	clust02page050
	clust02page046
C3	clust03page062
	clust03page063
C4	clust04page128
	clust04page125
	clust04page124
	clust04page129
	clust04page126
C5	clust05page002
	clust05page012
C6	clust06page088
	clust06page101
	clust06page109
	clust06page104
	clust06page094
C7	clust07page131
	clust07page130
	clust07page132
C8	clust08page003
	clust08page010
	clust08page006
	clust08page004
C9	clust09page031
	clust09page071
	clust09page018
	clust09page100
C10	clust10page083
	clust10page082
	clust10page081
	clust10page084
	clust10page058
	clust10page078
	clust10page077
	clust10page076
	clust10page075
	clust10page074
	clust10page017
	clust10page057
	clust10page085
	clust10page086
	clust10page070
	clust10page087
	clust10page047

C11	clust11page068
	clust11page066
	clust11page069
	clust11page120
	clust11page060
	clust11page065
C12	clust12page038
	clust12page039
	clust12page040
	clust12page037
	clust12page092
C13	clust13page121
	clust13page113
C14	clust14page021
	clust14page027
	clust14page055
	clust14page056
	clust14page030
C15	clust15page015
	clust15page014
	clust15page016
C16	clust16page112
	clust16page114
	clust16page115
	clust16page116
	clust16page117
	clust16page118
	clust16page119
	clust16page122
	clust16page123
	clust16page127
	clust16page133
	clust16page041
	clust16page036
	clust16page035
	clust16page034
	clust16page033
	clust16page032
	clust16page028
	clust16page025
	clust16page023
	clust16page022
	clust16page020
	clust16page019
	clust16page011
	clust16page009
	clust16page008
	clust16page007
	clust16page005
	clust16page001
	clust16page091
	clust16page090
	clust16page089
	clust16page080
	clust16page079
	clust16page073
	clust16page072
	clust16page067
	clust16page064
	clust16page061
	clust16page059
	clust16page054
	clust16page053

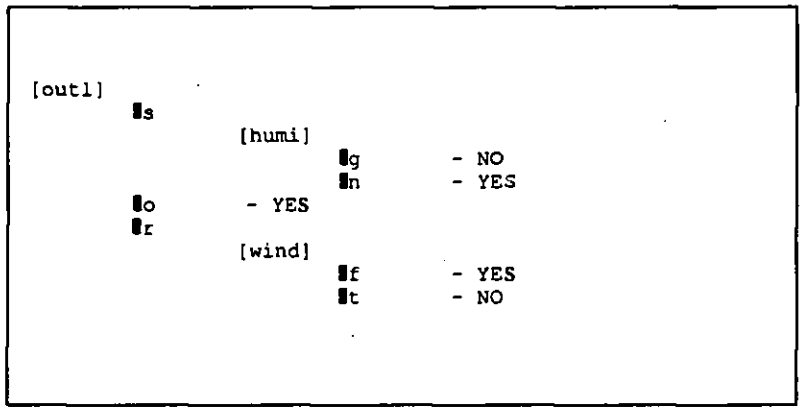
clust16page052
clust16page051
clust16page049
clust16page048
clust16page045
clust16page044
clust16page043
clust16page042
clust16page099
clust16page098
clust16page097
clust16page096
clust16page095
clust16page093
clust16page013
clust16page107
clust16page106
clust16page105
clust16page103
clust16page102
clust16page111
clust16page110
clust16page108

E.9 Quinlan's data set

The inductive learning algorithm ID3 (Quinlan 1986) was implemented without the windowing feature in the initial MLTutor prototype. In order to test the implementation of this algorithm, a standalone test was conducted using Quinlan's original test data. The results of this test, which match those of Quinlan, are shown below.

Example	outl	temp	humi	wind	play	*
1	s	h	g	f	no	*
2	s	h	g	t	no	*
3	o	h	g	f	yes	*
4	r	m	g	f	yes	*
5	r	c	n	f	yes	*
6	r	c	n	t	no	*
7	o	c	n	t	yes	*
8	s	m	g	f	no	*
9	s	c	n	f	yes	*
10	r	m	n	f	yes	*
11	s	m	n	t	yes	*
12	o	m	g	t	yes	*
13	o	h	n	f	yes	*
14	r	m	g	t	no	**

Quinlan' original test data.



The decision tree produced by the ID3 implementation in MLTutor.

E.10 Modified Quinlan's data set

In order to confirm the impact the SG-1, the data used to test the implementation of ID3 above was amended such that equal maximum information gain is achieved during the tree building process. The results of applying SG-1 to this data can be found below. the result of applying the ID3 algorithm to this data resulted in the same decision tree seen above.

Example	outl	out2	out3	temp	humi	wind	play	*
1	s	l	x	h	g	f	no	*
2	s	l	x	h	g	t	no	*
3	o	k	y	h	g	f	yes	*
4	r	d	z	m	g	f	yes	*
5	r	d	z	c	n	f	yes	*
6	r	d	z	c	n	t	no	*
7	o	k	y	c	n	t	yes	*
8	s	l	x	m	g	f	no	*
9	s	l	x	c	n	f	yes	*
10	r	d	z	m	n	f	yes	*
11	s	l	x	m	n	t	yes	*
12	o	k	y	m	g	t	yes	*
13	o	k	y	h	n	f	yes	*
14	r	d	z	m	g	t	no	**

Modified training data.

```
[outl]
        ▮s
                [humi]
                        ▮g      - NO
                        ▮n      - YES
        ▮o      - YES
        ▮r
                [wind]
                        ▮f      - YES
                        ▮t      - NO
[out2]
        ▮l
                [humi]
                        ▮g      - NO
                        ▮n      - YES
        ▮k      - YES
        ▮d
                [wind]
                        ▮f      - YES
                        ▮t      - NO
[out3]
        ▮x
                [humi]
                        ▮g      - NO
                        ▮n      - YES
        ▮y      - YES
        ▮z
                [wind]
                        ▮f      - YES
                        ▮t      - NO
```

SG-1 decision tree generated for the modified data.

Appendix F

F.1 Source code of the ID3 algorithm

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>

//#define DEBUG

extern unsigned _stklen = 50000U;

/*************************************************************************/
int id3(int file_code)
{
#define M1 15   // no of rows in learn.dat > no of items clustered
#define M2 150  // no of atts for ID3 > atts in attrib.dat                    //it was 150
before
#define M3 5    // length of string describing attribute                     //it was 4
before
#define M4 141  // no of attributes in attrib.dat + 1 (for EOS)      //it was 101 before

int data_load(char string[M1][M2][M3], int *, int *, char title[M2][M3]);
int check_all_positive(char string[M1][M2][M3], int, int);
int check_all_negative(char string[M1][M2][M3], int, int);
int get_diff_att_types(char valid[M1], char string[M1][M2][M3],
                                        char att_names[M1][M3], int, int);
int create_tree(char rule[M4], char avail_att[M2], FILE *ofp,
                                        FILE *nfp, FILE *pfp, char string[M1][M2][M3],
                                        char valid[M1], int, int, char title[M2][M3], int);
int not_all_same(char valid[M1], char string[M1][M2][M3], int attributes);

int attributes = 0, rows = 0, tab_cnt = 0;
char string[M1][M2][M3];
char title[M2][M3];
char valid[M1];
char avail_att[M2];
char rule[M4];
FILE *ofp, *pfp, *nfp;

if (data_load(string,&attributes,&rows,title) == 999)
                {
        //printf("load\n");
                return 0;
                }

//printf("%d\n", attributes);
//printf("%d\n", rows);

if (file_code == 1)
         {
         if ((ofp = fopen("d_tree.dat","w")) == NULL)
                {
                printf("File error : Cannot create output file TREE.DAT\n");
                return 0;
                }
          if ((pfp = fopen("d_pos.dat","w")) == NULL)
                {
                printf("File error : Cannot create output file POSITIVE.DAT\n");
                return 0;
                }
          if ((nfp = fopen("d_neg.dat","w")) == NULL)
                {
                printf("File error : Cannot create output file NEGATIVE.DAT\n");
                return 0;
                }
        }
else
         {
         if ((ofp = fopen("d_tree.dat","a")) == NULL)
                {
                printf("File error : Cannot create output file TREE.DAT\n");
                return 0;
                }
          if ((pfp = fopen("d_pos.dat","a")) == NULL)
                {
                printf("File error : Cannot create output file POSITIVE.DAT\n");
                return 0;
                }
          if ((nfp = fopen("d_neg.dat","a")) == NULL)
                {
                printf("File error : Cannot create output file NEGATIVE.DAT\n");
                return 0;
                }
        }

fprintf(pfp, "rule\n");
fprintf(nfp, "rule\n");
fprintf(ofp, "\n");
```

```
if (check_all_positive(string,attributes,rows))
                    {
                    fprintf(ofp,"HALT:all_positive\n");
                    fprintf(pfp,"HALT:all_positive\n");
                    fprintf(nfp,"HALT:all_positive\n");
                    fprintf(pfp, "rule_end\n");
                    fprintf(nfp, "rule_end\n");
                    fclose(ofp);
                    fclose(nfp);
                    fclose(pfp);
                    return 1;
                    }

if (check_all_negative(string,attributes,rows))
                    {
                    fprintf(ofp,"HALT:all_negative\n");
                    fprintf(pfp,"HALT:all_negative\n");
                    fprintf(nfp,"HALT:all_negative\n");
                    fprintf(pfp, "rule_end\n");
                    fprintf(nfp, "rule_end\n");
                    fclose(ofp);
                    fclose(nfp);
                    fclose(pfp);
                    return 1;
                    }

memset (valid, 42, rows);     // set to '*'
memset (avail_att, 42, M2);   // set to '*'
memset (rule, 45, M4);        // set to '-'

if (create_tree(rule, avail_att, ofp, nfp, pfp, string, valid, rows, attributes, title, tab_cnt) ==
999)
                    {
                    return 0;
                    }

fprintf(pfp, "rule_end\n");
fprintf(nfp, "rule_end\n");

fclose(ofp);
fclose(pfp);
fclose(nfp);

return 1;
}

/*************************************************************************/
int create_tree(char rule[M4], char avail_att[M2], FILE *ofp,
                                        FILE *nfp, FILE *pfp, char string[M1][M2][M3],
                                        char valid[M1], int rows, int attributes, char
title[M2][M3],
                                        int tab_cnt)
{
int get_diff_att_types(char valid[M1], char string[M1][M2][M3],
                                        char att_names[M1][M3], int, int);
int not_all_same(char valid[M1], char string[M1][M2][M3], int attributes);
int find_att(char avail_att[M2], char string[M1][M2][M3], char valid[M1],
                                         int attributes, int rows);

char att_names[M1][M3] = {" "};
char valid_2[M1];
char avail_att_2[M2];
char rule_2[M4];
int  j, l, i, ret, tot_diff_atts, att_no;

for (i=0;i<tab_cnt+tab_cnt;++i)
                    {
                    fprintf(ofp,"\t");
                    }

tab_cnt++;

if ((att_no = find_att(avail_att, string, valid, attributes, rows)) == 999)
                    {
        //printf("attno\n");
                    return 999;
                    }

rule[M4-1] = '\0';                  //make string
avail_att[M2-1] = '\0';             //make string
strcpy(avail_att_2, avail_att);
avail_att_2[att_no] = ' ';

fprintf(ofp, "[%s]\n", title[att_no]);

tot_diff_atts = get_diff_att_types(valid, string, att_names, att_no, rows);

for (j=0;j<tot_diff_atts;++j)
                    {

                    valid[M1-1] = '\0';
                    strcpy(valid_2,valid);

                    for (l=0;l<rows;++l)
```

```
                                        {
                                        if (strcmp(att_names[j],string[1][att_no]) != 0)
                                                {
                                                valid_2[1] = ' ';
                                                }
                                        }

                if ((ret = not_all_same(valid_2,string,attributes)) == 1)
                                        {
                                        for (i=0;i<tab_cnt+tab_cnt-1;++i)
                                                {
                                                fprintf(ofp,"\t");
                                                }

                                        fprintf(ofp,"%s\n",att_names[j]);
                                        rule[att_no-1] = att_names[j][0];
                                        strcpy(rule_2, rule);

                                        if (create_tree(rule_2, avail_att_2, ofp, nfp, pfp,
                                                string, valid_2, rows,
attributes,title,tab_cnt) == 999)
                                                {
                                                return 999;
                                                }
                                        }
                else
                                        {
                                        for (i=0;i<tab_cnt+tab_cnt-1;++i)
                                                {
                                                fprintf(ofp,"\t");
                                                }
                                        if (ret == 2)
                                                {
                                                fprintf(ofp,"%s\t -
YES\n",att_names[j]);
                                                rule[att_no-1] =
att_names[j][0];
                                                fprintf(pfp,"%s\n",rule);
                                                }
                                        else
                                                {
                                                fprintf(ofp,"%s\t -
NO\n",att_names[j]);
                                                rule[att_no-1] =
att_names[j][0];
                                                fprintf(nfp,"%s\n",rule);
                                                }
                                        }
                }
return 1;
}

/*****************************************************************************/
int find_att(char avail_att[M2], char string[M1][M2][M3], char valid[M1],
                                        int attributes, int rows)
{

int get_diff_att_types(char valid[M1], char string[M1][M2][M3],
                                        char att_names[M1][M3], int, int);
void disaster(int);

int i, j, l, y_tot = 0, n_tot = 0, y_tot_2, n_tot_2;
int tot_diff_atts;
int att_no = 0;
double max_inf_gain = -1.0;
double entropy, entropy_2, r_entropy_tot;
double att_entropy[M2];
char att_names[M1][M3] = {" "};
char valid_2[M1];

// CHOSE ONE OF THE FOLLOWING

// THIS IS THE MAX INFO GAIN

for (i=0;i<M2;++i)
                {
                att_entropy[i] = -2.0;
                }

// THIS IS THE MIN INFO GAIN

//for (i=0;i<M2;++i)
//              {
//              att_entropy[i] = 2.0;
//              }

// CHOSE ONE OF THE ABOVE

for (i=1;i<=M1;i++)
                {
                if (valid[i] == '*')
                                        {
                                        if (strcmp(string[i][attributes],"yes") == 0)
                                                ++y_tot;
```

```
                                        if (strcmp(string[i][attributes],"no") == 0)
                                                ++n_tot;
                                        }
                }

if (y_tot == 0 || n_tot == 0)
                entropy = 0.0;
else
                {
                entropy =  0.0 - ((y_tot/(double)(y_tot+n_tot))
        *log((y_tot/(double)(y_tot+n_tot))))
                                                -
((n_tot/(double)(y_tot+n_tot))*
        log((n_tot/(double)(y_tot+n_tot))));
                }

for (i=1;i<attributes;++i)
                {
                if (avail_att[i] == '*')
                                        {
                                        r_entropy_tot = 0.0;
                                        tot_diff_atts = get_diff_att_types(valid, string,
                                                        att_names, i, rows);
                                        for (j=0;j<tot_diff_atts;++j)
                                                {
                                                memset (valid_2, 32, M1);
                                                for (l=1;l<=rows;++l)
                                                        {
                                                        if
((strcmp(att_names[j],string[l][i]) == 0)
                && (valid[l] == '*'))
                valid_2[l] = '*';
                                                        }
                                                y_tot_2 = 0;
                                                n_tot_2 = 0;
                                                for (l=1;l<=M1;l++)
                                                        {
                                                        if
(valid_2[l] == '*')
                {
                if (strcmp(string[l][attributes],"yes") == 0)
                                        ++y_tot_2;
                if (strcmp(string[l][attributes],"no") == 0)
                                        ++n_tot_2;
                }
                                                        }
                                                if (n_tot_2 == 0 || y_tot_2 ==
0)
                                                        entropy_2 =
0.0;
                                                else
                                                        {
                                                        entropy_2 =
0.0 - ((y_tot_2/(double)(y_tot_2+n_tot_2))
                                        *log((y_tot_2/(double)(y_tot_2+n_tot_2))))
                                        - ((n_tot_2/(double)(y_tot_2+n_tot_2))
                                        *log((n_tot_2/(double)(y_tot_2+n_tot_2))));
                                                        }
                                                r_entropy_tot = r_entropy_tot +
(entropy_2
                * ((n_tot_2+y_tot_2)/(double)(n_tot+y_tot)));
                                                }
                                        att_entropy[i] = entropy - r_entropy_tot;
                                        }
                }
// CHOSE ONE OF THE FOLLOWING
// THIS IS THE MAX INFO GAIN
```

```
for (l=0;l<M2;++l)
                    {
                    if (att_entropy[l] >= max_inf_gain)
                                        {
                                        max_inf_gain = att_entropy[l];
                                        att_no = l;
                                        }
                    }

if (max_inf_gain == 0.0)
                    {
                    disaster(l);
                    return 999;
                    }

// THIS IS THE MIN INFO GAIN

//max_inf_gain = 1.99;

//for (l=0;l<M2;++l)
//                  {
//                  if (att_entropy[l] <= max_inf_gain)
//                                      {
//                                      max_inf_gain = att_entropy[l];
//                                      att_no = l;
//                                      }
//                  }

//if (max_inf_gain == 1.99)
//                  {
//                  disaster(l);
//                  return 999;
//                  }

// CHOOSE ONE OF THE ABOVE

return att_no;

}

/*************************************************************************/
int not_all_same(char valid[M1], char string[M1][M2][M3], int attributes)
{
int i, y_tot = 0, n_tot = 0;

for (i=0;i<M1;i++)
                    {
                    if (valid[i] == '*')
                                        {
                                        if (strcmp(string[i][attributes],"yes") == 0)
                                                            ++y_tot;
                                        if (strcmp(string[i][attributes],"no") == 0)
                                                            ++n_tot;
                                        }
                    }

if (n_tot == 0)
                    return 2;     /* all yes */
else if (y_tot == 0)
                    return 3;     /* all no  */
else
                    return 1;
}

/*************************************************************************/
int get_diff_att_types(char valid[M1], char string[M1][M2][M3],
                                                                        char
att_names[M1][M3], int att, int max_row)
{
int j,l,k;
char att_temp[M1][M3];

for(j=0;j<max_row;j++)
                    {
                    strcpy(att_names[j],string[j][att]);
                    }

for(l=0;l<j;++l)
                    {
                    if (valid[l] != '*')
                                         memset(att_names[l], 42, M3-1);
                    }

for(j=0;j<max_row;j++)
                    {
                    l=1;
                    for(l=l+j;l<max_row;l++)
                                        {
                                        if (strcmp(att_names[j],att_names[l]) == 0)
                                                            {
                                                            memset(att_names[l], 42, M3-1);
                                                            }
                                        }
                    }
```

```
for(l=0,k=0;l<j;l++)
                    {
                    if (att_names[l][0] != '*')
                                        {
                                        strcpy(att_temp[k],att_names[l]);
                                        k++;
                                        }
                    }

for(l=0;l<j;++l)
                    {
                    memset(att_names[l], 42, M3-1);
                    }

for(l=0;l<k;++l)
                    {
                    strcpy(att_names[l],att_temp[l]);
                    }

for(l=0,k=0;l<j;l++)
                    {
                    if (att_names[l][0] != '*')
                                        ++k;
                    }

return k;

}

/*************************************************************************/
int check_all_positive(char string[M1][M2][M3], int attributes, int rows)
{
int i;

for(i=0;i<rows;++i)
                    {
                    if (strcmp(string[i][attributes],"no") == 0)
                                        {
                                        return 0;
                                        }
                    }
return 1;
}

/*************************************************************************/
int check_all_negative(char string[M1][M2][M3], int attributes, int rows)
{
int i;

for(i=0;i<rows;++i)
                    {
                    if (strcmp(string[i][attributes],"yes") == 0)
                                        {
                                        return 0;
                                        }
                    }
return 1;
}

/*************************************************************************/
int data_load(char string[M1][M2][M3], int* a, int* b, char title[M2][M3])
{
char linebuff[20];
int k=0;

FILE *ifp;

if ((ifp = fopen("d_learn.dat","r")) == NULL)
                    {
                    printf("File error : Cannot open input file LEARN.DAT\n");
                    return 999;
                    }

do {
                    fscanf(ifp,"%s",title[k]);
          }while(title[k++][0] != '*');

do {
                    (*a)=0;

                    do     {
                                        fscanf(ifp,"%s",linebuff);
                                        strcpy(string[(*b)][(*a)],linebuff);
                                        (*a)++;
                                        }while(linebuff[0] != '*');

                    (*b)++;

          }while(linebuff[1] !='*');

*a = *a - 2;

fclose(ifp);
```

```
return 1;
}

/*************************************************************************/
void disaster(int i)
{

switch(i)
                {
                case 1: printf("** ID3 failure **\n");
                                                        //system("cls");
                                        //printf("\nA serious error has occured.\n\n");
                                        //printf("All output files may be corrupt.\n\n");
                                        //printf("Possible inconistancies or contradictory\n");
                                        //printf("input cases may be the cause.\n\n");
                                        //printf("\n\nPress any key");
                                        //getche();
                                        break;
                }
}

/*************************************************************************/
```

F.2 Source code of the SG-1 algorithm

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>

//#define DEBUG

extern unsigned _stklen = 500000U;

/*************************************************************************/
int id3(int file_code)
{
#define M1 15   // no of rows in learn.dat > no of items clustered
#define M2 150  // no of atts for ID3 > atts in attrib.dat                    //it was 150
before
#define M3 5    // length of string describing attribute                     //it was 4
before
#define M4 141  // no of attributes in attrib.dat + 1 (for EOS)        //it was 101 before

int data_load(char string[M1][M2][M3], int *, int *, char title[M2][M3]);
int check_all_positive(char string[M1][M2][M3], int, int);
int check_all_negative(char string[M1][M2][M3], int, int);
int get_diff_att_types(char valid[M1], char string[M1][M2][M3],
                                    char att_names[M1][M3], int, int);
int create_tree(char rule[M4], char avail_att[M2], FILE *ofp,
                                    FILE *nfp, FILE *pfp, char string[M1][M2][M3],
                                    char valid[M1], int, int, char title[M2][M3], int);
int not_all_same(char valid[M1], char string[M1][M2][M3], int attributes);

int attributes = 0, rows = 0, tab_cnt = -1;
char string[M1][M2][M3];
char title[M2][M3];
char valid[M1];
char avail_att[M2];
char rule[M4];
FILE *ofp, *pfp, *nfp;

if (data_load(string,&attributes,&rows,title) == 999)
                {
                //printf("load\n");
                return 0;
                }

//printf("%d\n", attributes);
//printf("%d\n", rows);

if (file_code == 1)
        {
        if ((ofp = fopen("d_tree.dat","w")) == NULL)
                {
                printf("File error : Cannot create output file TREE.DAT\n");
                return 0;
                }
         if ((pfp = fopen("d_pos.dat","w")) == NULL)
                {
                printf("File error : Cannot create output file POSITIVE.DAT\n");
                return 0;
                }
         if ((nfp = fopen("d_neg.dat","w")) == NULL)
                {
                printf("File error : Cannot create output file NEGATIVE.DAT\n");
                return 0;
                }
         }
else
         {
         if ((ofp = fopen("d_tree.dat","a")) == NULL)
                {
                printf("File error : Cannot create output file TREE.DAT\n");
                return 0;
                }
         if ((pfp = fopen("d_pos.dat","a")) == NULL)
                {
                printf("File error : Cannot create output file POSITIVE.DAT\n");
                return 0;
                }
         if ((nfp = fopen("d_neg.dat","a")) == NULL)
                {
                printf("File error : Cannot create output file NEGATIVE.DAT\n");
                return 0;
                }
        }

fprintf(pfp, "rule\n");
fprintf(nfp, "rule\n");
```

```
fprintf(ofp, "\n");

if (check_all_positive(string,attributes,rows))
                {
                fprintf(ofp,"HALT:all_positive\n");
                fprintf(pfp,"HALT:all_positive\n");
                fprintf(nfp,"HALT:all_positive\n");
                fprintf(pfp, "rule_end\n");
                fprintf(nfp, "rule_end\n");
                fclose(ofp);
                fclose(nfp);
                fclose(pfp);
                return 1;
                }

if (check_all_negative(string,attributes,rows))
                {
                fprintf(ofp,"HALT:all_negative\n");
                fprintf(pfp,"HALT:all_negative\n");
                fprintf(nfp,"HALT:all_negative\n");
                fprintf(pfp, "rule_end\n");
                fprintf(nfp, "rule_end\n");
                fclose(ofp);
                fclose(nfp);
                fclose(pfp);
                return 1;
                }

memset (valid, 42, rows);     // set to '*'
memset (avail_att, 42, M2);   // set to '*'
memset (rule, 45, M4);        // set to '-'

if (create_tree(rule, avail_att, ofp, nfp, pfp, string, valid, rows, attributes, title, tab_cnt) ==
999)
                {
                return 0;
                }

fprintf(pfp, "rule_end\n");
fprintf(nfp, "rule_end\n");

fclose(ofp);
fclose(pfp);
fclose(nfp);

return 1;
}

/*************************************************************************/
int create_tree(char rule[M4],
                                        char avail_att[M2],
                                        FILE *ofp,
                                        FILE *nfp,
                                        FILE *pfp,
                                        char string[M1][M2][M3],
                                        char valid[M1],
                                        int rows, int attributes,
                                        char title[M2][M3],
                                        int tab_cnt)
        {
        int get_diff_att_types(char valid[M1],
                                                                char string[M1][M2][M3],
                                                                char att_names[M1][M3],
                                                                int,
                                                                int);

        int not_all_same(char valid[M1],
                                                char string[M1][M2][M3],
                                                int attributes);

        int find_att(char avail_att[M2],
                                        char string[M1][M2][M3],
                                        char valid[M1],
                                        int attributes,
                                        int rows,
                                        int function_code,
                                        int which_best);

        char att_names[M1][M3] = {" "};
        char valid_2[M1];
        char avail_att_2[M2];
        char rule_2[M4];
        char rule_work[M4];
        int j;
        int l;
        int i;
        int ret;
        int tot_diff_atts;
        int att_no;
        int function_code;
        int equal_best;
        int which_best;
        int for_each_rule;
```

```
//      for (i=0;i<tab_cnt+tab_cnt;++i)
//              {
//              fprintf(ofp,"\t");
//              }

        tab_cnt++;

        which_best = 999;
        function_code = 1;

        if ((equal_best = find_att(avail_att, string, valid, attributes, rows, function_code,
which_best)) == 999)
                {
                return 999;
                }

//      printf("Equal best : %d \n", equal_best); getche();

        for (for_each_rule = 1; for_each_rule <= equal_best; for_each_rule++)
                {
//              printf("for each rule : %d \n", for_each_rule); getche();

                function_code = 3;
                which_best = for_each_rule;

                if ((att_no = find_att(avail_att, string, valid, attributes, rows, function_code,
which_best)) == 999)
                        {
                        return 999;
                        }

                rule[M4-1] = '\0';                      //make string
                avail_att[M2-1] = '\0';                 //make string
                strcpy(rule_work, rule);
                strcpy(avail_att_2, avail_att);

                avail_att_2[att_no] = ' ';

                for (i=0;i<tab_cnt+tab_cnt;++i)
                        {
                        fprintf(ofp,"\t");
                        }

//              tab_cnt++;

                fprintf(ofp, "[%s]\n", title[att_no]);

                tot_diff_atts = get_diff_att_types(valid, string, att_names, att_no, rows);

                for (j=0;j<tot_diff_atts;++j)
                        {
                        valid[M1-1] = '\0';
                        strcpy(valid_2,valid);
                        for (l=0;l<rows;++l)
                                {
                                if (strcmp(att_names[j],string[l][att_no]) != 0)
                                        {
                                        valid_2[l] = ' ';
                                        }
                                }
                        if ((ret = not_all_same(valid_2,string,attributes)) == 1)
                                {
                                for (i=0;i<tab_cnt+tab_cnt+1;++i)
                                        {
                                        fprintf(ofp,"\t");
                                        }
                                fprintf(ofp,"□%s\n",att_names[j]);
                                rule_work[att_no-1] = att_names[j][0];
                                strcpy(rule_2, rule_work);
                                if (create_tree(rule_2, avail_att_2, ofp, nfp, pfp,
                                                string, valid_2, rows, attributes,title,tab_cnt) ==
999)
                                        {
                                        return 999;
                                        }
                                }
                        else
                                {
                                for (i=0;i<tab_cnt+tab_cnt+1;++i)
                                        {
                                        fprintf(ofp,"\t");
                                        }
                                if (ret == 2)
                                        {
                                        fprintf(ofp,"□%s\t - YES\n",att_names[j]);
                                        rule_work[att_no-1] = att_names[j][0];
                                        fprintf(pfp,"%s\n",rule_work);
                                        }
                                else
                                        {
                                        fprintf(ofp,"□%s\t - NO\n",att_names[j]);
                                        rule_work[att_no-1] = att_names[j][0];
                                        fprintf(nfp,"%s\n",rule_work);
                                        }
```

```
                                }
                        }
                } // end for each of the best

        return 1;
        }

/*****************************************************************/
int find_att(char avail_att[M2],
                                char string[M1][M2][M3],
                                char valid[M1],
                                int attributes,
                                int rows,
                                int function_code,
                                int which_best)
        {

        int get_diff_att_types(char valid[M1],
                                                                char string[M1][M2][M3],
                                                                char att_names[M1][M3],
                                                                int,
                                                                int);

        void disaster(int);

        int i, j, l, y_tot = 0, n_tot = 0, y_tot_2, n_tot_2;
        int tot_diff_atts;
        int att_no = 0;
        double max_inf_gain = -1.0;
        double entropy, entropy_2, r_entropy_tot;
        double att_entropy[M2];
        char att_names[M1][M3] = {" "};
        char valid_2[M1];
        int equal_best;

// CHOSE ONE OF THE FOLLOWING

// THIS IS THE MAX INFO GAIN

        for (i=0;i<M2;++i)
                {
                att_entropy[i] = -2.0;
                }

// THIS IS THE MIN INFO GAIN

// for (i=0;i<M2;++i)
//              {
//              att_entropy[i] = 2.0;
//              }

// CHOSE ONE OF THE ABOVE

        for (i=1;i<=M1;i++)
                {
                if (valid[i] == '*')
                        {
                        if (strcmp(string[i][attributes],"yes") == 0)
                                {
                                ++y_tot;
                                }
                        if (strcmp(string[i][attributes],"no") == 0)
                                {
                                ++n_tot;
                                }
                        }
                }

        if (y_tot == 0 || n_tot == 0)
                {
                entropy = 0.0;
                }
        else
                {
                entropy =  0.0 - ((y_tot/(double)(y_tot+n_tot))
                                                                * log((y_tot/(double)(y_tot+n_tot))))
                                                                - ((n_tot/(double)(y_tot+n_tot))
                                                                *
log((n_tot/(double)(y_tot+n_tot))));
                }

        for (i=1;i<attributes;++i)
                {
                if (avail_att[i] == '*')
                        {
                        r_entropy_tot = 0.0;
                        tot_diff_atts = get_diff_att_types(valid, string, att_names, i, rows);
                        for (j=0;j<tot_diff_atts;++j)
                                {
                                memset (valid_2, 32, M1);
                                for (l=1;l<=rows;++l)
                                        {
                                        if ((strcmp(att_names[j],string[l][i]) == 0) && (valid[l] ==
'*'))
                                                {
```

```
                                        valid_2[l] = '*';
                                        }
                                }
                        y_tot_2 = 0;
                        n_tot_2 = 0;
                        for (l=1;l<=M1;l++)
                                {
                                if (valid_2[l] == '*')
                                        {
                                        if (strcmp(string[l][attributes],"yes") == 0)
                                                {
                                                ++y_tot_2;
                                                }
                                        if (strcmp(string[l][attributes],"no") == 0)
                                                {
                                                ++n_tot_2;
                                                }
                                        }
                                }
                        if (n_tot_2 == 0 || y_tot_2 == 0)
                                {
                                entropy_2 = 0.0;
                                }
                        else
                                {
                                entropy_2 =  0.0 - ((y_tot_2/(double)(y_tot_2+n_tot_2))
log((y_tot_2/(double)(y_tot_2+n_tot_2))))
-
((n_tot_2/(double)(y_tot_2+n_tot_2))
log((n_tot_2/(double)(y_tot_2+n_tot_2))));
                                }
                        r_entropy_tot = r_entropy_tot + ((entropy_2
((n_tot_2+y_tot_2)/(double)(n_tot+y_tot)));
                        }
                        att_entropy[i] = entropy - r_entropy_tot;
                }
        }

//CHOSE ONE OF THE FOLLOWING

// THIS IS THE MAX INFO GAIN

        equal_best = 0;

        for (l=0;l<M2;++l)
                {
                if (att_entropy[l] >= max_inf_gain)
                        {
//                      printf("Att entropy : %f \n", att_entropy[l]); getche();
//                      if ((att_entropy[l] == max_inf_gain) && (max_inf_gain >= 0.0))
//                              {
//                              equal_best++;
//                              }
                        max_inf_gain = att_entropy[l];
                        att_no = l;
                        }
                }

        if (function_code == 1)
                {
                for (l=0;l<M2;++l)
                        {
                        if (att_entropy[l] >= max_inf_gain)
                                {
//                              printf("Att entropy : %f \n", att_entropy[l]); getche();
                                equal_best++;
                                }
                        }
                }

//      printf("Max info gain : %f \n", max_inf_gain); getche();

        if (function_code == 3)
                {
                equal_best = 0;
                for (l=0;l<M2;++l)
                        {
//                      printf("Att entropy : %f \n", att_entropy[l]); getche();
                        if (att_entropy[l] >= max_inf_gain)
                                {
                                att_no = l;
                                equal_best++;
                                }
                        if (which_best == equal_best)
                                {
                                break;
                                }
                        }
                }

//      if (equal_best > 0)
```

```
//              {
//              printf("Equal best : %d %f %d \n", equal_best, max_inf_gain, att_no); getche();
//              }

        if (max_inf_gain == 0.0)
                {
                disaster(1); return 999;
                }

// THIS IS THE MIN INFO GAIN

// max_inf_gain = 1.99;

// for (l=0;l<M2;++l)
//         {
//         if (att_entropy[l] <= max_inf_gain)
//            {
//                 max_inf_gain = att_entropy[l];
//                 att_no = l;
//                 }
//         }

// if (max_inf_gain == 1.99)
//         {
//         disaster(1);
//         return 999;
//         }

// CHOOSE ONE OF THE ABOVE

        if (function_code == 0)
              , {
                return att_no;
                }

        if (function_code == 1)
                {
                return equal_best;
                }

        if (function_code == 3)
                {
                return att_no;
                }

        }

/*************************************************************************/
int not_all_same(char valid[Ml], char string[M1][M2][M3], int attributes)
{
int i, y_tot = 0, n_tot = 0;

for (i=0;i<Ml;i++)
                {
                if (valid[i] == '*')
                                        {
                                        if (strcmp(string[i][attributes],"yes") == 0)
                                                ++y_tot;
                                        if (strcmp(string[i][attributes],"no") == 0)
                                                ++n_tot;
                                        }
                }

if (n_tot == 0)
                return 2;    /* all yes */
else if (y_tot == 0)
                return 3;    /* all no  */
else
                return 1;
}

/*************************************************************************/
int get_diff_att_types(char valid[M1], char string[M1][M2][M3],
                                                                        char
att_names[M1][M3], int att, int max_row)
{
int j,l,k;
char att_temp[M1][M3];

for(j=0;j<max_row;j++)
                {
                strcpy(att_names[j],string[j][att]);
                }

for(l=0;l<j;++l)
                {
                if (valid[l] != '*')
                                        memset(att_names[l], 42, M3-1);
                }

for(j=0;j<max_row;j++)
                {
                l=1;
                for(l=l+j;l<max_row;l++)
```

```
                                        {
                                        if (strcmp(att_names[j],att_names[l]) == 0)
                                                {
                                                memset(att_names[l], 42, M3-1);
                                                }
                                        }
                    }

for(l=0,k=0;l<j;l++)
                    {
                    if (att_names[l][0] != '*')
                                        {
                                        strcpy(att_temp[k],att_names[l]);
                                        k++;
                                        }
                    }

for(l=0;l<j;++l)
                    {
                    memset(att_names[l], 42, M3-1);
                    }

for(l=0;l<k;++l)
                    {
                    strcpy(att_names[l],att_temp[l]);
                    }

for(l=0,k=0;l<j;l++)
                    {
                    if (att_names[l][0] != '*')
                                        ++k;
                    }

return k;

}

/*************************************************************************/
int check_all_positive(char string[M1][M2][M3], int attributes, int rows)
{
int i;

for(i=0;i<rows;++i)
                    {
                    if (strcmp(string[i][attributes],"no") == 0)
                                        {
                                        return 0;
                                        }
                    }
return 1;
}

/*************************************************************************/
int check_all_negative(char string[M1][M2][M3], int attributes, int rows)
{
int i;

for(i=0;i<rows;++i)
                    {
                    if (strcmp(string[i][attributes],"yes") == 0)
                                        {
                                        return 0;
                                        }
                    }
return 1;
}

/*************************************************************************/
int data_load(char string[M1][M2][M3], int* a, int* b, char title[M2][M3])
{
char linebuff[20];
int k=0;

FILE *ifp;

if ((ifp = fopen("d_learn.dat","r")) == NULL)
                    {
                    printf("File error : Cannot open input file LEARN.DAT\n");
                    return 999;
                    }

do {
                    fscanf(ifp,"%s",title[k]);
        }while(title[k++][0] != '*');

do {
                    (*a)=0;

                    do    {
                                fscanf(ifp,"%s",linebuff);
                                strcpy(string[(*b)][(*a)],linebuff);
                                (*a)++;
                                }while(linebuff[0] != '*');
```

```
                (*b)++;

        }while(linebuff[1] !='*');

*a = *a - 2;

fclose(ifp);

return 1;
}

/*************************************************************************/
void disaster(int i)
{

switch(i)
                {
                case 1: printf("** ID3 failure **\n");
                                                        //system("cls");
                                        //printf("\nA serious error has occured.\n\n");
                                        //printf("All output files may be corrupt.\n\n");
                                        //printf("Possible inconistancies or contradictory\n");
                                        //printf("input cases may be the cause.\n\n");
                                        //printf("\n\nPress any key");
                                        //getche();
                                        break;
                }
}

/*************************************************************************/
```

F.3 Source code of the conceptual clustering algorithm

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>

#include "MLTClust.h"

//#define DEBUG

#define LIMIT1 10                                       // how many records input ie pages in input1.dat
#define LIMIT2 (((LIMIT1 * LIMIT1) - LIMIT1) / 2)       // required array size

int clust()
{

// Function prototype definitions

int find_cluster_diameter(char *, cluster_record *, attribute_record_out *);
int nearly_central(int, char *, cluster_record *, attribute_record_out *);
int find_distance_1(attribute_data, attribute_data);
int sort_function_1(cluster_record *, cluster_record *);
int sort_function_2(attribute_record_out *, attribute_record_out *);
attribute_data find_subj_attribute_vector(char *, FILE *);
void new_name(char *, int *);

// Local variables

FILE *ifp, *ofp, *afp;
int i, j, k, max, biggest, new_clust = 1, llnes;
int last_page_num, latest, actual;
int cur_weight, max_weight, heaviest;
char buffer[8], temp_a[8], temp_b[8], last_page[8];
attribute_record_out attribute_rec_out[LIMIT1];
cluster_record sort_array[LIMIT2];

// Open the input and output attribute record files.

if ((ifp = fopen("d_input1.dat","r")) == NULL)
     {
        printf("File error : Cannot open input file INPUT1.DAT\n");
        return 0;
        }

if ((ofp = fopen("d_output.dat","w")) == NULL)
        {
        printf("File error : Cannot create output file OUTPUT.DAT\n");
        return 0;
        }

if ((afp = fopen("d_attrib.dat","r")) == NULL)
        {
        printf("File error : Cannot open attribute file ATTRIB.DAT\n");
        return 0;
        }

// Read the subj identifiers into memory and access the attribute
// descriptions from the subj attribute database.

fseek(ifp, -(LIMIT1 * 9), 2);

for (i=0; i<LIMIT1; ++i)
        {
        fscanf(ifp, "%s", attribute_rec_out[i].attribute_record_in.subj_id);
        attribute_rec_out[i].attribute_record_in.subj_attributes =
        find_subj_attribute_vector(attribute_rec_out[i].attribute_record_in.subj_id, afp);
        strcpy(attribute_rec_out[i].assigned_cluster_name, "DUMMY");
        attribute_rec_out[i].weight = i+1;
        if (i==LIMIT1-1)
                {
                strcpy(last_page,attribute_rec_out[i].attribute_record_in.subj_id);
                //printf("Last page id is : %s ", last_page);
                last_page_num = atoi(last_page+5);
                //printf("Last page id is : %d ", last_page_num);
                }
        }

// Load the sort array with pairs of unordered input points. Add the
// distance between the pairs to the array.

k = 0;
for (i=0; i<LIMIT1; ++i)
        {
        for (j=i+1; j<LIMIT1; ++j)
                {
                sort_array[k].input_case_a = i;
                sort_array[k].input_case_b = j;
                sort_array[k].distance_a_to_b =
```

```
                find_distance_1(attribute_rec_out[i].attribute_record_in.subj_attributes,
                        attribute_rec_out[j].attribute_record_in.subj_attributes);
                ++k;
                }
        }

// Sort the sort array

qsort(sort_array, LIMIT2, sizeof(cluster_record), (int(*)(const void *,const void
*))sort_function_1);

// Perform the clustering

for (k=0; k<LIMIT2; k++)
        {
        if ((strcmp(attribute_rec_out[sort_array[k].input_case_a].assigned_cluster_name, "DUMMY") ==
0)
        && (strcmp(attribute_rec_out[sort_array[k].input_case_b].assigned_cluster_name, "DUMMY") ==
0))
                {
                // put both into a new cluster
                new_name(buffer, &new_clust);
                strcpy(attribute_rec_out[sort_array[k].input_case_a].assigned_cluster_name, buffer);
                strcpy(attribute_rec_out[sort_array[k].input_case_b].assigned_cluster_name, buffer);
                }
        if ((strcmp(attribute_rec_out[sort_array[k].input_case_a].assigned_cluster_name, "DUMMY") ==
0)
        && (strcmp(attribute_rec_out[sort_array[k].input_case_b].assigned_cluster_name, "DUMMY") !=
0))
                {
                // one in cluster one not so put the other in the same cluster
                strcpy(attribute_rec_out[sort_array[k].input_case_a].assigned_cluster_name,
                attribute_rec_out[sort_array[k].input_case_b].assigned_cluster_name);
                }
        if ((strcmp(attribute_rec_out[sort_array[k].input_case_a].assigned_cluster_name, "DUMMY") !=
0)
        && (strcmp(attribute_rec_out[sort_array[k].input_case_b].assigned_cluster_name, "DUMMY") ==
0))
                {
                // one in cluster one not so put the other in the same cluster
                strcpy(attribute_rec_out[sort_array[k].input_case_b].assigned_cluster_name,
                attribute_rec_out[sort_array[k].input_case_a].assigned_cluster_name);
                }
        if ((strcmp(attribute_rec_out[sort_array[k].input_case_a].assigned_cluster_name, "DUMMY") !=
0)
        && (strcmp(attribute_rec_out[sort_array[k].input_case_b].assigned_cluster_name, "DUMMY") !=
0)
        && (strcmp(attribute_rec_out[sort_array[k].input_case_a].assigned_cluster_name,
           attribute_rec_out[sort_array[k].input_case_b].assigned_cluster_name) != 0)
        && (nearly_central(sort_array[k].input_case_a,
                        attribute_rec_out[sort_array[k].input_case_a].assigned_cluster_name,
                        sort_array,
                        attribute_rec_out))
        && (nearly_central(sort_array[k].input_case_b,
                        attribute_rec_out[sort_array[k].input_case_b].assigned_cluster_name,
                        sort_array,
                        attribute_rec_out))
        && ((float)sort_array[k].distance_a_to_b <
(((float)(find_cluster_diameter(attribute_rec_out[sort_array[k].input_case_a].assigned_cluster_name,
sort_array, attribute_rec_out) +

find_cluster_diameter(attribute_rec_out[sort_array[k].input_case_b].assigned_cluster_name,
sort_array, attribute_rec_out)) / 2))))
                {
                // both in clusters but different ones
                new_name(buffer, &new_clust);

strcpy(temp_a,attribute_rec_out[sort_array[k].input_case_a].assigned_cluster_name);

strcpy(temp_b,attribute_rec_out[sort_array[k].input_case_b].assigned_cluster_name);
                for (i=0; i<LIMIT1; ++i)
                        {
                        if ((strcmp(attribute_rec_out[i].assigned_cluster_name,temp_a) == 0)
                        || (strcmp(attribute_rec_out[i].assigned_cluster_name,temp_b) == 0))
                                strcpy(attribute_rec_out[i].assigned_cluster_name,buffer);
                        }
                }
        }

// Sort the output records

qsort(attribute_rec_out, LIMIT1, sizeof(attribute_record_out), (int(*)(const void *,const void
*))sort_function_2);

// Write the clustered records to the output file

fprintf(ofp, "clust00subj000 a001 a002 a003 a004 a005 a006 a007 a008 a009 a010 ");
fprintf(ofp, "a011 a012 a013 a014 a015 a016 a017 a018 a019 a020 a021 a022 a023 a024 ");
fprintf(ofp, "a025 a026 a027 a028 a029 a030 a031 a032 a033 a034 a035 a036 a037 a038 ");
fprintf(ofp, "a039 a040 a041 a042 a043 a044 a045 a046 a047 a048 a049 a050 a051 a052 ");
fprintf(ofp, "a053 a054 a055 a056 a057 a058 a059 a060 a061 a062 a063 a064 a065 a066 ");
fprintf(ofp, "a067 a068 a069 a070 a071 a072 a073 a074 a075 a076 a077 a078 a079 a080 ");
fprintf(ofp, "a081 a082 a083 a084 a085 a086 a087 a088 a089 a090 a091 a092 a093 a094 ");
fprintf(ofp, "a095 a096 a097 a098 a099 a100 a101 a102 a103 a104 a105 a106 a107 a108 ");
```

```
fprintf(ofp, "a109 a110 a111 a112 a113 a114 a115 a116 a117 a118 a119 a120 a121 a122 ");
fprintf(ofp, "a123 a124 a125 a126 a127 a128 a129 a130 a131 a132 a133 a134 a135 a136 ");
fprintf(ofp, "a137 a138 a139 a140      *\n");

strcpy(temp_a,"1234567");

k = 0;          // the current cluster number (reuse k)
j = 0;          // the number of lines in cluster (reuse j)
lines = 0;      // the number of lines written for cluster
max = 0;        // the number of lines in the largest cluster so far
biggest = 0;    // the number of the biggest cluster so far
actual = 0;     // the number of the current cluster
latest = 0;     // the cluster containing the most recent page
cur_weight = 0; // the current cluster weight
max_weight = 0; // the largest cluster weight
heaviest = 0;   // the cluster with greatest weight

for (i=0; i<LIMITI; ++i)
        {
        if (strcmp(attribute_rec_out[i].assigned_cluster_name,temp_a) != 0)
                {
                strcpy(temp_a, attribute_rec_out[i].assigned_cluster_name);

                actual++;               // increment for each cluster processed

                cur_weight = 0;         // reset for a new cluster

                lines = 0;              // reset for new cluster
                }

        lines++;                        // increment for each line written

        if (lines > max)
                {
                max = lines;                    // update max
                biggest = actual;               // store for return
                }

        cur_weight = cur_weight + attribute_rec_out[i].weight;

        if (cur_weight > max_weight)
                {
                max_weight = cur_weight;        // update max
                heaviest = actual;              // save heaviest so far
                }

        //printf("Cluster weight %d \n", cur_weight);
        //getche();

        if (strcmp(attribute_rec_out[i].attribute_record_in.subj_id,last_page) == 0)
                {
                latest = actual;        // latest is the cluster position NOT no
                }

        fprintf(ofp, "%s%s    %d    %d    %d    %d    %d    %d    ",
                  attribute_rec_out[i].assigned_cluster_name,
                  attribute_rec_out[i].attribute_record_in.subj_id,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute01,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute02,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute03,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute04,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute05,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute06);

        fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    %d    ",
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute07,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute08,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute09,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute10,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute11,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute12,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute13,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute14);

        fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute15,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute16,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute17,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute18,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute19,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute20,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute21);

        fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute22,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute23,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute24,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute25,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute26,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute27,
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute28);

        fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
                  attribute_rec_out[i].attribute_record_in.subj_attributes.attribute29,
```

```
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute30,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute31,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute32,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute33,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute34,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute35);

fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute36,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute37,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute38,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute39,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute40,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute41,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute42);

fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute43,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute44,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute45,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute46,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute47,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute48,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute49);

fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute50,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute51,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute52,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute53,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute54,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute55,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute56);

fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute57,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute58,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute59,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute60,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute61,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute62,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute63);

fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute64,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute65,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute66,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute67,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute68,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute69,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute70);

fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute71,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute72,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute73,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute74,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute75,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute76,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute77);

fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute78,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute79,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute80,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute81,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute82,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute83,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute84);

fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute85,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute86,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute87,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute88,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute89,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute90,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute91);

fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute92,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute93,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute94,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute95,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute96,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute97,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute98);

fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute99,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute100,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute101,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute102,
            attribute_rec_out[i].attribute_record_in.subj_attributes.attribute103,
```

```
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute104,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute105);

        fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute106,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute107,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute108,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute109,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute110,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute111,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute112);

        fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute113,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute114,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute115,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute116,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute117,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute118,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute119);

        fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute120,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute121,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute122,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute123,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute124,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute125,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute126);

        fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d    ",
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute127,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute128,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute129,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute130,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute131,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute132,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute133);

        fprintf(ofp, "%d    %d    %d    %d    %d    %d    %d xxx ",
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute134,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute135,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute136,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute137,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute138,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute139,
                attribute_rec_out[i].attribute_record_in.subj_attributes.attribute140);

        if (i == LIMIT1 - 1)
                {
                fprintf(ofp, "**\n");
                }
        else
                {
                fprintf(ofp, "*\n");
                }

        }

fclose(ifp);
fclose(ofp);
fclose(afp);

//printf("Weights : %d %d \n", heaviest, max_weight);
//printf("Largest : %d %d \n", biggest, max);
//printf("Latest  : %d \n", latest);

//getche();

//return actual;      // this is the number of clusters
//return biggest;     // this is the position of the biggest cluster
//return latest;      // this is the position of the cluster with the last page in it
return heaviest;      // this is the position of the heaviest cluster
}

// Fuction to determine if a record is nearly central in a cluster.

int nearly_central(int record_no, char * cluster_name, cluster_record dist_array[],
attribute_record_out output_records[])
{
int find_cluster_diameter(char *, cluster_record *, attribute_record_out *);

int diameter, k;
float delta;

diameter = find_cluster_diameter(cluster_name, dist_array, output_records);
delta = (float)diameter * 0.666;

for (k=0; k<LIMIT2; ++k)
                {
                if ((((dist_array[k].input_case_a == record_no)
                && (strcmp(output_records[dist_array[k].input_case_a].assigned_cluster_name,
                        output_records[dist_array[k].input_case_b].assigned_cluster_name) == 0))
```

```
                    || ((dist_array[k].input_case_b == record_no)
                    && (strcmp(output_records[dist_array[k].input_case_b].assigned_cluster_name,
                            output_records[dist_array[k].input_case_a].assigned_cluster_name) == 0)))

                    && ((float)dist_array[k].distance_a_to_b > delta))
                                        return(0);
                    }
return(1);
}

// Find diameter of the requested cluster.

int find_cluster_diameter(char * cluster_name, cluster_record dist_array[], attribute_record_out
output_records[])
{
int k, max_dist = -1;

for (k=0; k<LIMIT2; k++)
                    {
                    if ((strcmp(output_records[dist_array[k].input_case_a].assigned_cluster_name,
cluster_name) == 0)
                    && (strcmp(output_records[dist_array[k].input_case_b].assigned_cluster_name,
cluster_name) == 0)
                    && (dist_array[k].distance_a_to_b > max_dist))
                                        max_dist = dist_array[k].distance_a_to_b;
                    }
return(max_dist);
}

// Generate a new name for a cluster.

void new_name(char *buffer, int *value)
{
strcpy(buffer, "clust");
sprintf(buffer+5, "%02d", *value);
(*value)++;
}

// This function searches the attribute database and returns the
// attribute description for the supplied subj id.

attribute_data find_subj_attribute_vector(char * search_key, FILE *fp)
{
attribute_file_record afr;
char att_rec[198];              // change to length of the attribute record
int pos;

pos = atoi(search_key+4);
fseek(fp, (pos * 199), 0);      // change to match length of attribute record + 1

fscanf(fp, "%s", att_rec);

att_rec[147] = '\0';
afr.subj_attributes.attribute140 = atoi(att_rec+146);
att_rec[146] = '\0';
afr.subj_attributes.attribute139 = atoi(att_rec+145);
att_rec[145] = '\0';
afr.subj_attributes.attribute138 = atoi(att_rec+144);
att_rec[144] = '\0';
afr.subj_attributes.attribute137 = atoi(att_rec+143);
att_rec[143] = '\0';
afr.subj_attributes.attribute136 = atoi(att_rec+142);
att_rec[142] = '\0';
afr.subj_attributes.attribute135 = atoi(att_rec+141);
att_rec[141] = '\0';
afr.subj_attributes.attribute134 = atoi(att_rec+140);
att_rec[140] = '\0';
afr.subj_attributes.attribute133 = atoi(att_rec+139);
att_rec[139] = '\0';
afr.subj_attributes.attribute132 = atoi(att_rec+138);
att_rec[138] = '\0';
afr.subj_attributes.attribute131 = atoi(att_rec+137);
att_rec[137] = '\0';
afr.subj_attributes.attribute130 = atoi(att_rec+136);
att_rec[136] = '\0';
afr.subj_attributes.attribute129 = atoi(att_rec+135);
att_rec[135] = '\0';
afr.subj_attributes.attribute128 = atoi(att_rec+134);
att_rec[134] = '\0';
afr.subj_attributes.attribute127 = atoi(att_rec+133);
att_rec[133] = '\0';
afr.subj_attributes.attribute126 = atoi(att_rec+132);
att_rec[132] = '\0';
afr.subj_attributes.attribute125 = atoi(att_rec+131);
att_rec[131] = '\0';
afr.subj_attributes.attribute124 = atoi(att_rec+130);
att_rec[130] = '\0';
afr.subj_attributes.attribute123 = atoi(att_rec+129);
att_rec[129] = '\0';
afr.subj_attributes.attribute122 = atoi(att_rec+128);
att_rec[128] = '\0';
afr.subj_attributes.attribute121 = atoi(att_rec+127);
att_rec[127] = '\0';
```

```
afr.subj_attributes.attribute120 = atoi(att_rec+126);
att_rec[126] = '\0';
afr.subj_attributes.attribute119 = atoi(att_rec+125);
att_rec[125] = '\0';
afr.subj_attributes.attribute118 = atoi(att_rec+124);
att_rec[124] = '\0';
afr.subj_attributes.attribute117 = atoi(att_rec+123);
att_rec[123] = '\0';
afr.subj_attributes.attribute116 = atoi(att_rec+122);
att_rec[122] = '\0';
afr.subj_attributes.attribute115 = atoi(att_rec+121);
att_rec[121] = '\0';
afr.subj_attributes.attribute114 = atoi(att_rec+120);
att_rec[120] = '\0';
afr.subj_attributes.attribute113 = atoi(att_rec+119);
att_rec[119] = '\0';
afr.subj_attributes.attribute112 = atoi(att_rec+118);
att_rec[118] = '\0';
afr.subj_attributes.attribute111 = atoi(att_rec+117);
att_rec[117] = '\0';
afr.subj_attributes.attribute110 = atoi(att_rec+116);
att_rec[116] = '\0';
afr.subj_attributes.attribute109 = atoi(att_rec+115);
att_rec[115] = '\0';
afr.subj_attributes.attribute108 = atoi(att_rec+114);
att_rec[114] = '\0';
afr.subj_attributes.attribute107 = atoi(att_rec+113);
att_rec[113] = '\0';
afr.subj_attributes.attribute106 = atoi(att_rec+112);
att_rec[112] = '\0';
afr.subj_attributes.attribute105 = atoi(att_rec+111);
att_rec[111] = '\0';
afr.subj_attributes.attribute104 = atoi(att_rec+110);
att_rec[110] = '\0';
afr.subj_attributes.attribute103 = atoi(att_rec+109);
att_rec[109] = '\0';
afr.subj_attributes.attribute102 = atoi(att_rec+108);
att_rec[108] = '\0';
afr.subj_attributes.attribute101 = atoi(att_rec+107);
att_rec[107] = '\0';
afr.subj_attributes.attribute100 = atoi(att_rec+106);
att_rec[106] = '\0';
afr.subj_attributes.attribute99 = atoi(att_rec+105);
att_rec[105] = '\0';
afr.subj_attributes.attribute98 = atoi(att_rec+104);
att_rec[104] = '\0';
afr.subj_attributes.attribute97 = atoi(att_rec+103);
att_rec[103] = '\0';
afr.subj_attributes.attribute96 = atoi(att_rec+102);
att_rec[102] = '\0';
afr.subj_attributes.attribute95 = atoi(att_rec+101);
att_rec[101] = '\0';
afr.subj_attributes.attribute94 = atoi(att_rec+100);
att_rec[100] = '\0';
afr.subj_attributes.attribute93 = atoi(att_rec+99);
att_rec[99] = '\0';
afr.subj_attributes.attribute92 = atoi(att_rec+98);
att_rec[98] = '\0';
afr.subj_attributes.attribute91 = atoi(att_rec+97);
att_rec[97] = '\0';
afr.subj_attributes.attribute90 = atoi(att_rec+96);
att_rec[96] = '\0';
afr.subj_attributes.attribute89 = atoi(att_rec+95);
att_rec[95] = '\0';
afr.subj_attributes.attribute88 = atoi(att_rec+94);
att_rec[94] = '\0';
afr.subj_attributes.attribute87 = atoi(att_rec+93);
att_rec[93] = '\0';
afr.subj_attributes.attribute86 = atoi(att_rec+92);
att_rec[92] = '\0';
afr.subj_attributes.attribute85 = atoi(att_rec+91);
att_rec[91] = '\0';
afr.subj_attributes.attribute84 = atoi(att_rec+90);
att_rec[90] = '\0';
afr.subj_attributes.attribute83 = atoi(att_rec+89);
att_rec[89] = '\0';
afr.subj_attributes.attribute82 = atoi(att_rec+88);
att_rec[88] = '\0';
afr.subj_attributes.attribute81 = atoi(att_rec+87);
att_rec[87] = '\0';
afr.subj_attributes.attribute80 = atoi(att_rec+86);
att_rec[86] = '\0';
afr.subj_attributes.attribute79 = atoi(att_rec+85);
att_rec[85] = '\0';
afr.subj_attributes.attribute78 = atoi(att_rec+84);
att_rec[84] = '\0';
afr.subj_attributes.attribute77 = atoi(att_rec+83);
att_rec[83] = '\0';
afr.subj_attributes.attribute76 = atoi(att_rec+82);
att_rec[82] = '\0';
afr.subj_attributes.attribute75 = atoi(att_rec+81);
att_rec[81] = '\0';
afr.subj_attributes.attribute74 = atoi(att_rec+80);
att_rec[80] = '\0';
```

```
afr.subj_attributes.attribute73 = atoi(att_rec+79);
att_rec[79] = '\0';
afr.subj_attributes.attribute72 = atoi(att_rec+78);
att_rec[78] = '\0';
afr.subj_attributes.attribute71 = atoi(att_rec+77);
att_rec[77] = '\0';
afr.subj_attributes.attribute70 = atoi(att_rec+76);
att_rec[76] = '\0';
afr.subj_attributes.attribute69 = atoi(att_rec+75);
att_rec[75] = '\0';
afr.subj_attributes.attribute68 = atoi(att_rec+74);
att_rec[74] = '\0';
afr.subj_attributes.attribute67 = atoi(att_rec+73);
att_rec[73] = '\0';
afr.subj_attributes.attribute66 = atoi(att_rec+72);
att_rec[72] = '\0';
afr.subj_attributes.attribute65 = atoi(att_rec+71);
att_rec[71] = '\0';
afr.subj_attributes.attribute64 = atoi(att_rec+70);
att_rec[70] = '\0';
afr.subj_attributes.attribute63 = atoi(att_rec+69);
att_rec[69] = '\0';
afr.subj_attributes.attribute62 = atoi(att_rec+68);
att_rec[68] = '\0';
afr.subj_attributes.attribute61 = atoi(att_rec+67);
att_rec[67] = '\0';
afr.subj_attributes.attribute60 = atoi(att_rec+66);
att_rec[66] = '\0';
afr.subj_attributes.attribute59 = atoi(att_rec+65);
att_rec[65] = '\0';
afr.subj_attributes.attribute58 = atoi(att_rec+64);
att_rec[64] = '\0';
afr.subj_attributes.attribute57 = atoi(att_rec+63);
att_rec[63] = '\0';
afr.subj_attributes.attribute56 = atoi(att_rec+62);
att_rec[62] = '\0';
afr.subj_attributes.attribute55 = atoi(att_rec+61);
att_rec[61] = '\0';
afr.subj_attributes.attribute54 = atoi(att_rec+60);
att_rec[60] = '\0';
afr.subj_attributes.attribute53 = atoi(att_rec+59);
att_rec[59] = '\0';
afr.subj_attributes.attribute52 = atoi(att_rec+58);
att_rec[58] = '\0';
afr.subj_attributes.attribute51 = atoi(att_rec+57);
att_rec[57] = '\0';
afr.subj_attributes.attribute50 = atoi(att_rec+56);
att_rec[56] = '\0';
afr.subj_attributes.attribute49 = atoi(att_rec+55);
att_rec[55] = '\0';
afr.subj_attributes.attribute48 = atoi(att_rec+54);
att_rec[54] = '\0';
afr.subj_attributes.attribute47 = atoi(att_rec+53);
att_rec[53] = '\0';
afr.subj_attributes.attribute46 = atoi(att_rec+52);
att_rec[52] = '\0';
afr.subj_attributes.attribute45 = atoi(att_rec+51);
att_rec[51] = '\0';
afr.subj_attributes.attribute44 = atoi(att_rec+50);
att_rec[50] = '\0';
afr.subj_attributes.attribute43 = atoi(att_rec+49);
att_rec[49] = '\0';
afr.subj_attributes.attribute42 = atoi(att_rec+48);
att_rec[48] = '\0';
afr.subj_attributes.attribute41 = atoi(att_rec+47);
att_rec[47] = '\0';
afr.subj_attributes.attribute40 = atoi(att_rec+46);
att_rec[46] = '\0';
afr.subj_attributes.attribute39 = atoi(att_rec+45);
att_rec[45] = '\0';
afr.subj_attributes.attribute38 = atoi(att_rec+44);
att_rec[44] = '\0';
afr.subj_attributes.attribute37 = atoi(att_rec+43);
att_rec[43] = '\0';
afr.subj_attributes.attribute36 = atoi(att_rec+42);
att_rec[42] = '\0';
afr.subj_attributes.attribute35 = atoi(att_rec+41);
att_rec[41] = '\0';
afr.subj_attributes.attribute34 = atoi(att_rec+40);
att_rec[40] = '\0';
afr.subj_attributes.attribute33 = atoi(att_rec+39);
att_rec[39] = '\0';
afr.subj_attributes.attribute32 = atoi(att_rec+38);
att_rec[38] = '\0';
afr.subj_attributes.attribute31 = atoi(att_rec+37);
att_rec[37] = '\0';
afr.subj_attributes.attribute30 = atoi(att_rec+36);
att_rec[36] = '\0';
afr.subj_attributes.attribute29 = atoi(att_rec+35);
att_rec[35] = '\0';
afr.subj_attributes.attribute28 = atoi(att_rec+34);
att_rec[34] = '\0';
afr.subj_attributes.attribute27 = atoi(att_rec+33);
att_rec[33] = '\0';
```

```
afr.subj_attributes.attribute26 = atoi(att_rec+32);
att_rec[32] = '\0';
afr.subj_attributes.attribute25 = atoi(att_rec+31);
att_rec[31] = '\0';
afr.subj_attributes.attribute24 = atoi(att_rec+30);
att_rec[30] = '\0';
afr.subj_attributes.attribute23 = atoi(att_rec+29);
att_rec[29] = '\0';
afr.subj_attributes.attribute22 = atoi(att_rec+28);
att_rec[28] = '\0';
afr.subj_attributes.attribute21 = atoi(att_rec+27);
att_rec[27] = '\0';
afr.subj_attributes.attribute20 = atoi(att_rec+26);
att_rec[26] = '\0';
afr.subj_attributes.attribute19 = atoi(att_rec+25);
att_rec[25] = '\0';
afr.subj_attributes.attribute18 = atoi(att_rec+24);
att_rec[24] = '\0';
afr.subj_attributes.attribute17 = atoi(att_rec+23);
att_rec[23] = '\0';
afr.subj_attributes.attribute16 = atoi(att_rec+22);
att_rec[22] = '\0';
afr.subj_attributes.attribute15 = atoi(att_rec+21);
att_rec[21] = '\0';
afr.subj_attributes.attribute14 = atoi(att_rec+20);
att_rec[20] = '\0';
afr.subj_attributes.attribute13 = atoi(att_rec+19);
att_rec[19] = '\0';
afr.subj_attributes.attribute12 = atoi(att_rec+18);
att_rec[18] = '\0';
afr.subj_attributes.attribute11 = atoi(att_rec+17);
att_rec[17] = '\0';
afr.subj_attributes.attribute10 = atoi(att_rec+16);
att_rec[16] = '\0';
afr.subj_attributes.attribute09 = atoi(att_rec+15);
att_rec[15] = '\0';
afr.subj_attributes.attribute08 = atoi(att_rec+14);
att_rec[14] = '\0';
afr.subj_attributes.attribute07 = atoi(att_rec+13);
att_rec[13] = '\0';
afr.subj_attributes.attribute06 = atoi(att_rec+12);
att_rec[12] = '\0';
afr.subj_attributes.attribute05 = atoi(att_rec+11);
att_rec[11] = '\0';
afr.subj_attributes.attribute04 = atoi(att_rec+10);
att_rec[10] = '\0';
afr.subj_attributes.attribute03 = atoi(att_rec+9);
att_rec[9] = '\0';
afr.subj_attributes.attribute02 = atoi(att_rec+8);
att_rec[8] = '\0';
afr.subj_attributes.attribute01 = atoi(att_rec+7);
att_rec[7] = '\0';
strcpy(afr.subj_id, att_rec);

if (strcmp(afr.subj_id, search_key) == 0)
                return (afr.subj_attributes);

printf ("Attribute not found for key!!! %s %d \n", search_key, pos);

return (afr.subj_attributes); // to prevent compiler warning only
}

// This is the cluster QSORT sort function.

int sort_function_1(cluster_record *first, cluster_record *second)
{
        if (first->distance_a_to_b < second->distance_a_to_b)
                return (-1);
        else if (first->distance_a_to_b > second->distance_a_to_b)
                return (+1);
        else
                return (0);
}

// This is the output QSORT sort function.

int sort_function_2(attribute_record_out *first, attribute_record_out *second)
{
        return(strcmp(first->assigned_cluster_name, second->assigned_cluster_name));
}

// This function finds the distance between two attribute records. The
// pseudo metric employed is the number of differing attribute values.

int find_distance_1(attribute_data rec01, attribute_data rec02)
{
int distance = 0;

if (rec01.attribute01 != rec02.attribute01)
                ++distance;

if (rec01.attribute02 != rec02.attribute02)
                ++distance;
```

```
if (rec01.attribute03 != rec02.attribute03)
                    ++distance;

if (rec01.attribute04 != rec02.attribute04)
                    ++distance;

if (rec01.attribute05 != rec02.attribute05)
                    ++distance;

if (rec01.attribute06 != rec02.attribute06)
                    ++distance;

if (rec01.attribute07 != rec02.attribute07)
                    ++distance;

if (rec01.attribute08 != rec02.attribute08)
                    ++distance;

if (rec01.attribute09 != rec02.attribute09)
                    ++distance;

if (rec01.attribute10 != rec02.attribute10)
                    ++distance;

if (rec01.attribute11 != rec02.attribute11)
                    ++distance;

if (rec01.attribute12 != rec02.attribute12)
                    ++distance;

if (rec01.attribute13 != rec02.attribute13)
                    ++distance;

if (rec01.attribute14 != rec02.attribute14)
                    ++distance;

if (rec01.attribute15 != rec02.attribute15)
                    ++distance;

if (rec01.attribute16 != rec02.attribute16)
                    ++distance;

if (rec01.attribute17 != rec02.attribute17)
                    ++distance;

if (rec01.attribute18 != rec02.attribute18)
                    ++distance;

if (rec01.attribute19 != rec02.attribute19)
                    ++distance;

if (rec01.attribute20 != rec02.attribute20)
                    ++distance;

if (rec01.attribute21 != rec02.attribute21)
                    ++distance;

if (rec01.attribute22 != rec02.attribute22)
                    ++distance;

if (rec01.attribute23 != rec02.attribute23)
                    ++distance;

if (rec01.attribute24 != rec02.attribute24)
                    ++distance;

if (rec01.attribute25 != rec02.attribute25)
                    ++distance;

if (rec01.attribute26 != rec02.attribute26)
                    ++distance;

if (rec01.attribute27 != rec02.attribute27)
                    ++distance;

if (rec01.attribute28 != rec02.attribute28)
                    ++distance;

if (rec01.attribute29 != rec02.attribute29)
                    ++distance;

if (rec01.attribute30 != rec02.attribute30)
                    ++distance;

if (rec01.attribute31 != rec02.attribute31)
                    ++distance;

if (rec01.attribute32 != rec02.attribute32)
                    ++distance;

if (rec01.attribute33 != rec02.attribute33)
                    ++distance;

if (rec01.attribute34 != rec02.attribute34)
```

```
                    ++distance;
if (rec01.attribute35 != rec02.attribute35)
                    ++distance;
if (rec01.attribute36 != rec02.attribute36)
                    ++distance;
if (rec01.attribute37 != rec02.attribute37)
                    ++distance;
if (rec01.attribute38 != rec02.attribute38)
                    ++distance;
if (rec01.attribute39 != rec02.attribute39)
                    ++distance;
if (rec01.attribute40 != rec02.attribute40)
                    ++distance;
if (rec01.attribute41 != rec02.attribute41)
                    ++distance;
if (rec01.attribute42 != rec02.attribute42)
                    ++distance;
if (rec01.attribute43 != rec02.attribute43)
                    ++distance;
if (rec01.attribute44 != rec02.attribute44)
                    ++distance;
if (rec01.attribute45 != rec02.attribute45)
                    ++distance;
if (rec01.attribute46 != rec02.attribute46)
                    ++distance;
if (rec01.attribute47 != rec02.attribute47)
                    ++distance;
if (rec01.attribute48 != rec02.attribute48)
                    ++distance;
if (rec01.attribute49 != rec02.attribute49)
                    ++distance;
if (rec01.attribute50 != rec02.attribute50)
                    ++distance;
if (rec01.attribute51 != rec02.attribute51)
                    ++distance;
if (rec01.attribute52 != rec02.attribute52)
                    ++distance;
if (rec01.attribute53 != rec02.attribute53)
                    ++distance;
if (rec01.attribute54 != rec02.attribute54)
                    ++distance;
if (rec01.attribute55 != rec02.attribute55)
                    ++distance;
if (rec01.attribute56 != rec02.attribute56)
                    ++distance;
if (rec01.attribute57 != rec02.attribute57)
                    ++distance;
if (rec01.attribute58 != rec02.attribute58)
                    ++distance;
if (rec01.attribute59 != rec02.attribute59)
                    ++distance;
if (rec01.attribute60 != rec02.attribute60)
                    ++distance;
if (rec01.attribute61 != rec02.attribute61)
                    ++distance;
if (rec01.attribute62 != rec02.attribute62)
                    ++distance;
if (rec01.attribute63 != rec02.attribute63)
                    ++distance;
if (rec01.attribute64 != rec02.attribute64)
                    ++distance;
if (rec01.attribute65 != rec02.attribute65)
                    ++distance;
```

```
if (rec01.attribute66 != rec02.attribute66)
                    ++distance;

if (rec01.attribute67 != rec02.attribute67)
                    ++distance;

if (rec01.attribute68 != rec02.attribute68)
                    ++distance;

if (rec01.attribute69 != rec02.attribute69)
                    ++distance;

if (rec01.attribute70 != rec02.attribute70)
                    ++distance;

if (rec01.attribute71 != rec02.attribute71)
                    ++distance;

if (rec01.attribute72 != rec02.attribute72)
                    ++distance;

if (rec01.attribute73 != rec02.attribute73)
                    ++distance;

if (rec01.attribute74 != rec02.attribute74)
                    ++distance;

if (rec01.attribute75 != rec02.attribute75)
                    ++distance;

if (rec01.attribute76 != rec02.attribute76)
                    ++distance;

if (rec01.attribute77 != rec02.attribute77)
                    ++distance;

if (rec01.attribute78 != rec02.attribute78)
                    ++distance;

if (rec01.attribute79 != rec02.attribute79)
                    ++distance;

if (rec01.attribute80 != rec02.attribute80)
                    ++distance;

if (rec01.attribute81 != rec02.attribute81)
                    ++distance;

if (rec01.attribute82 != rec02.attribute82)
                    ++distance;

if (rec01.attribute83 != rec02.attribute83)
                    ++distance;

if (rec01.attribute84 != rec02.attribute84)
                    ++distance;

if (rec01.attribute85 != rec02.attribute85)
                    ++distance;

if (rec01.attribute86 != rec02.attribute86)
                    ++distance;

if (rec01.attribute87 != rec02.attribute87)
                    ++distance;

if (rec01.attribute88 != rec02.attribute88)
                    ++distance;

if (rec01.attribute89 != rec02.attribute89)
                    ++distance;

if (rec01.attribute90 != rec02.attribute90)
                    ++distance;

if (rec01.attribute91 != rec02.attribute91)
                    ++distance;

if (rec01.attribute92 != rec02.attribute92)
                    ++distance;

if (rec01.attribute93 != rec02.attribute93)
                    ++distance;

if (rec01.attribute94 != rec02.attribute94)
                    ++distance;

if (rec01.attribute95 != rec02.attribute95)
                    ++distance;

if (rec01.attribute96 != rec02.attribute96)
                    ++distance;
```

```
if (rec01.attribute97 != rec02.attribute97)
                    ++distance;

if (rec01.attribute98 != rec02.attribute98)
                    ++distance;

if (rec01.attribute99 != rec02.attribute99)
                    ++distance;

if (rec01.attribute100 != rec02.attribute100)
                    ++distance;

if (rec01.attribute101 != rec02.attribute101)
                    ++distance;

if (rec01.attribute102 != rec02.attribute102)
                    ++distance;

if (rec01.attribute103 != rec02.attribute103)
                    ++distance;

if (rec01.attribute104 != rec02.attribute104)
                    ++distance;

if (rec01.attribute105 != rec02.attribute105)
                    ++distance;

if (rec01.attribute106 != rec02.attribute106)
                    ++distance;

if (rec01.attribute107 != rec02.attribute107)
                    ++distance;

if (rec01.attribute108 != rec02.attribute108)
                    ++distance;

if (rec01.attribute109 != rec02.attribute109)
                    ++distance;

if (rec01.attribute110 != rec02.attribute110)
                    ++distance;

if (rec01.attribute111 != rec02.attribute111)
                    ++distance;

if (rec01.attribute112 != rec02.attribute112)
                    ++distance;

if (rec01.attribute113 != rec02.attribute113)
                    ++distance;

if (rec01.attribute114 != rec02.attribute114)
                    ++distance;

if (rec01.attribute115 != rec02.attribute115)
                    ++distance;

if (rec01.attribute116 != rec02.attribute116)
                    ++distance;

if (rec01.attribute117 != rec02.attribute117)
                    ++distance;

if (rec01.attribute118 != rec02.attribute118)
                    ++distance;

if (rec01.attribute119 != rec02.attribute119)
                    ++distance;

if (rec01.attribute120 != rec02.attribute120)
                    ++distance;

if (rec01.attribute121 != rec02.attribute121)
                    ++distance;

if (rec01.attribute122 != rec02.attribute122)
                    ++distance;

if (rec01.attribute123 != rec02.attribute123)
                    ++distance;

if (rec01.attribute124 != rec02.attribute124)
                    ++distance;

if (rec01.attribute125 != rec02.attribute125)
                    ++distance;

if (rec01.attribute126 != rec02.attribute126)
                    ++distance;

if (rec01.attribute127 != rec02.attribute127)
                    ++distance;

if (rec01.attribute128 != rec02.attribute128)
```

```
                    ++distance;

if (rec01.attribute129 != rec02.attribute129)
                    ++distance;

if (rec01.attribute130 != rec02.attribute130)
                    ++distance;

if (rec01.attribute131 != rec02.attribute131)
                    ++distance;

if (rec01.attribute132 != rec02.attribute132)
                    ++distance;

if (rec01.attribute133 != rec02.attribute133)
                    ++distance;

if (rec01.attribute134 != rec02.attribute134)
                    ++distance;

if (rec01.attribute135 != rec02.attribute135)
                    ++distance;

if (rec01.attribute136 != rec02.attribute136)
                    ++distance;

if (rec01.attribute137 != rec02.attribute137)
                    ++distance;

if (rec01.attribute138 != rec02.attribute138)
                    ++distance;

if (rec01.attribute139 != rec02.attribute139)
                    ++distance;

if (rec01.attribute140 != rec02.attribute140)
                    ++distance;

return distance;
}
```