Developing a design methodology for the construction of hypertext and hypermedia, with particular reference to hypertext electronic prospectuses

Thesis

UNRESTRICTED

Robert Urquhart Kemp, BA, MSc.

Developing a design methodology for the construction of hypertext and hypermedia, with particular reference to hypertext electronic prospectuses

Degree of Doctor of Philosophy

Information Science

31st May 1997

Queen Margaret College, Edinburgh

AWARDING BODY:
THE OPEN UNIVERSITY

Volume 1 of 2

Date of award: 29th September 1997

Abstract

Use and development of hypertext-based documents is becoming more widespread in both industry and the academic world. This has obvious implications for the design of hypertext documents. The hypertext designer has been relatively ignored until recently, with attention largely focused on the quality of the hypertext rather than support for the designer. Recent hypertext design methodologies, such as that described by Isakowitz et al. (1995) have made a useful contribution, but are oriented towards designers with a background in computing science and related professions.

This research addresses this problem by the development of a design methodology which is intended to be accessible to the general author. The design methodology was based on three sources of data: a taxonomy of existing design guidance, including a range of principles and guidelines and previous design methodologies for hypertext; hypertext versions of a higher education college prospectus, and a case study of a CD-ROM higher education prospectus.

This material was assembled and synthesised to produce a provisional design methodology that is positioned between existing design methodologies such as Relationship Management Methodology (Isakowitz et al 1995) and Object-Oriented Hypermedia Design Method (Schwabe et al 1995), which are influenced by software engineering and database design concepts, and other less formal descriptions of the hypertext design process. The design methodology supports and encourages iterative methods of working, and includes supporting documentation and pro formas designed to encourage a thorough approach to hypertext design.

Contents

Chapter 1

1.1 Research context and objectives [1]

1.1.1 Disciplinary issues [2]

1.2 Original contribution [2]

1.3 Methodological approaches [3]

1.3.1 The methodological context [3]

1.3.2 The method chosen - a triangulation approach [3]

1.4 Structure of this thesis [4]

Chapter 2

2.1 Introduction [6]

2.2 Hypertext and hypermedia [6]

2.2.1 Definitions and history [7]

2.2.2 Advantages and disadvantages of hypertext [15]

2.2.3 Hypertext features: links, nodes, structure and navigation [18]

2.3 Hypertext electronic prospectuses [25]

2. 3.1 Paper Prospectuses [25]

2.3.2 Electronic prospectuses: a definition [29]

2.3.3 History [29]

2.3.4 Current Position [31]

2.4 Design: the broader context [31]

2.4.1 Design in general: problems and solutions [32]

2.4.2 The systems approach in more detail [36]

2.5 The hypertext design process [40

2.5.1 To what extent are hypertext design problems hypertext-specific? [40]

2.5.2 What generalisations can be made about hypertext design? [41]

2.5.3 Why hypertext design is difficult: a problem-oriented exploration [44]

2.6 Hypertext design solutions: models, methodologies, guidelines [66]

2.7 Conclusions [68]

Chapter 3

3.1 Aims: what this research seeks to do [70]

3.1.1 Overall research aims [70]

3.2 What areas of knowledge are relevant? [71]

3.3 What we already know: implications of the literature review [71]

3.3.1 Why a hypertext design methodology is worthwhile [73]

3.4 Research design (1): what do we need to make a methodology? [75]

3.4.1 Theoretical context [76]

3.5 Research design (2) what are the likely sources of information? [78]

3.5.1 Information about existing design methodologies: what do design methodologies typically consist of? [79]

3.5.2 What information do you need to make a design methodology? [82]

3.5.3 So what might a hypertext design methodology consist of? [84]

3.6 Research design (3) what are the best methods for getting the required information? [86]

3.6.1 Choosing research methods: a review of alternatives [86]

3.6.2 The methods chosen [89]

3.7 Research design (4): How does one construct a design methodology from this information? [90]

3.7.1 Relationship of the data sources [90]

3.7.2 Putting together the design methodology [90]

3.8 Problems and limitations: the moving target: issues of time and change [92]

3.8.1 The role of data [92]

Chapter 4

4.1 Rationale: the role of the taxonomy in the overall study [93]

4.1.1 The nature, role & value of design guidance [94]

4.1.2 Summary: design guidance and design methodology [104]

4.2 Methods and procedure [106]

4.2.1 Method [106]

4.2.2 Procedure [107]

4.3. Results [128]

4.3.1 Problems [128]

4.4 Implications for a design methodology [130]

Chapter 5

5.1 Introduction [131]

5.1.1 Relationship to other parts of the research [131]

5.2 Method [132]

5.2.1 Methodological issues [133]

5.3 The electronic prospectus design process [133]

5.3.1 Introduction [133]

5.3.2 Design of the first prototype [134]

5.4 Later work [150]

5.5 Implications for the design methodology [150]

Chapter 6

6.1 Purpose of the case study [151]

6.2 Research method [151]

6.2.1 The value of case study research [151]

6.2.2 Other case study considerations [154]

6.2.3 The case study method in this research [154]

6.3 The case study procedure [157]

6.3.1 Case study protocol [157]

6.3.2 Data gathering techniques [158]

6.3.3 Choice of sites [158]

6.3.4 Participants [159]

6.3.5 Data gathering [159]

6.3.6 The analysis [160]

6.4 Results [164]

6.4.1 Interpretation and answers to Level 2 Questions [164]

6.5 Interpretation of results [175]

6.5.1 General [175]

6.5.2 Some reflections and criticisms [175]

Chapter 7

7.1 Introduction [183]

7.2 Rationale [183]

7.2.1 The problem area [184]

7.2.2 What do we mean by design methodology? [184]

7.2.3 The philosophy behind this methodology [186]

7.3. Method [200]

7.3.1 The data [201]

7.3.2 Procedure [202]

7.4. The results: The proposed design methodology [203]

7.4.1 The procedural phases in detail [203]

Chapter 8

8.1 Introduction [212]

8.2 Contribution to knowledge [212]

8.3 Further research [216]

8.3.1 Improving and extending the design methodology through testing and evaluation [216]

8.3.2 Extending and reclassifying the taxonomy [217]

8.3.3 Further case studies [217]

8.3.4 Development of the case study protocol [217]

Bibliography [219]

Appendix 1: Guidelines [249]

Appendix 2: Electronic prospectus design [273]

Appendix 3: Case study [305]

Appendix 4: Hypertext Authoring Design Methodology [359)

Acknowledgements

I would like to thank my supervisors, Jim Herring, Kathy Buckner and Steve Draper for their advice and guidance over the last few years.

Thanks are also due to my colleagues in the Communication Studies Department at Queen Margaret College for their general support and good humour, and especially Lizzie Davenport and Mark Percival for their perceptive comments at crucial points.

My family and friends have been an essential source of encouragement, tolerance, humour and understanding, for which I am deeply grateful.

List of figures

Figure 2.1	Classification of structure type: linear [21]
Figure 2.2	Classification of structure type: side excursion [21]
Figure 2.3	Classification of structure type: loop to beginning [21]
Figure 2.4	Classification of structure type: basic grid [22]
Figure 2.5	Classification of structure type: pure hierarchical [22]
Figure 2.6	Classification of structure type: lattices or hierarchical with cross references [22]
Figure 5.1	Information seeking functions [139]
Figure 6.1	Course information module structure [169]
Figure 6.2	Course structure [170]
Figure 7.1	M01: HADM design stages - linear [205]
Figure 7.2	M02: HADM design stages - asynchronous [206]

List of tables and diagrams

Table 4.1	Classification based on Avison and Fitzgerald [118-119]
Table 4.2	Modified classification [120-121]
Table 4.3	Final classification [122-23]
Diagram D01	Linked unstructured pattern note data [391]
Diagram D03	Structured links based on pattern note data (D01) [392]
Diagram D04	A hierarchical prospectus structure [393]

1: Introduction

1.1 Research context and objectives

Computers and their corollary, digital information, are becoming all-pervasive in today's society. It often seems that there is no end to the tasks that can be done by computer, or to the information that can be represented digitally. This research is concerned with one very small part of this expansion, a particular way of representing information digitally for a specific use: the design of hypertext documents. It explores the use of design methodologies in facilitating the development of such hypertext documents, examining their application for higher education prospectuses, and proposing a design methodology suitable for the design of these and similar hypertext documents.

Designing hypertext documents poses a number of problems, which have been outlined in some detail in the literature (see Chapter 2). As interactive computer systems, hypertext documents have many characteristics common to such systems, and similarly common design problems. It is suggested that hypertext documents also pose certain particular problems which stem from their novel status as a technology bridging the gap between the traditional world of print, text and books, and the newer realm of interactive systems. For the designer, this may make the cognitive leap to imagined design solutions unfeasibly large. Whereas many design situations involve the presentation of variations of relatively tried and tested solutions to the client and user in a context where familiarity with the artefacts involved means they have some reasonable conception of what is possible, here there is no such repertoire of existing artefacts to stimulate the elicitation of requirements. It is certainly the case that hypertext documents to date have met with mixed success, and it is suggested by a number of authorities that one reason for this is the quality of design. The principal objective of this research was therefore to create a design methodology which could ease the task of creating such documents both for the designer and the user, but concentrating in particular on the needs of the designer.

1.1.1 DISCIPLINARY ISSUES

Hypermedia's attributes, functions and contexts of use make it of interest to workers in a number of different disciplines. For instance hypertext has been studied within the disciplines of computer science, human computer interaction, information science and psychology. In this it parallels the study of text, a legitimate subject for many disciplines. It is not the intention of this author to appropriate hypertext as a subject for any particular discipline. However, the question as to which discipline or disciplines provides the most appropriate paradigm for an investigation into the chosen topic must be considered. This research has been seen by the author as loosely situated within the field of Information Science, which in itself can be characterised as a discipline of Social Science. This is appropriate in that the central emphasis of the research is on how people in a particular social (work) context might best conduct a particular task. However, it must be recognised that other disciplines have an interest in hypermedia, and their perspectives may be both legitimate and useful. In particular, hypertext design is often seen as a part of the discipline of Human-Computer Interaction (HCI), and this research draws extensively on the philosophies and practices associated with HCI.

1.2 Original contribution

The research examines a relatively un-investigated area and in doing so makes several contributions to knowledge. It explores the possibilities for a hypertext design methodology, originally conceived for electronic prospectuses, but capable of generalisation to other related small to medium sized information systems, and presents an example of such a design methodology. It contains a compilation and taxonomy of relevant principles, guidelines, and methodologies for the design of hypertext information systems. It employs an adaptation of Yin's (1994) method for case study research in the form of a case study protocol and accompanying interview schedule which may be used for the investigation of a range of hypertext design situations. The case study research itself provides a contribution to studies of the design process in action.

1.3 Methodological approaches

1.3.1 THE METHODOLOGICAL CONTEXT

The study takes its methodological perspective from work in the fields of software design research, information systems and human-computer interaction which suggests that theory-based research is at present of limited value in such areas (Carroll 1991, Landauer 1991). Traditional methods of scientific inquiry are of limited use because the study of information systems design and human computer interaction is essentially a moving target. The technology concerned, and the way in which it is used is constantly changing. This makes it difficult if not impossible to make generalisations that have any meaning.

1.3.2 THE METHOD CHOSEN - A TRIANGULATION APPROACH

Three sources of data were chosen. First of all the existing design advice was examined. This was initially expected to serve simply as a source of reference for the actual document design, and later for the development of a methodology, but the difficulties of following and making sense of the various forms of advice suggested that a systematised collection of the material would in itself be useful. Guidelines, principles and methodologies were therefore collected and collated. This material was organised and classified into a taxonomy of design guidance. The material on principles and guidelines was then edited and condensed into a set covering all aspects of the hypertext design process.

Secondly, it was decided to build a small-scale prototype electronic prospectus. It was originally intended that this provide the basis of a generalisable methodology, but instead this proved to have the principal function of acting as a sensitising device for this researcher, providing a preunderstanding of the subject and drawing attention to significant or problematic areas in the design of such documents. It also proved instrumental in extending knowledge of the key user groups.

Thirdly, an examination of practice was conducted. This took the form of a case study of the design and construction of an electronic prospectus published in CD-ROM form.

Finally, the three strands of the project were used as the data for the compilation of an electronic prospectus development methodology. The compilation process consisted of adaptation of suitable existing hypertext design methodologies augmented by items selected from the condensed taxonomy of design guidance. This material was supplemented on the basis of observations in the course of the author's own work, and from the case studies.

1.4 Structure of this thesis

This thesis continues with a literature review examining the existing situation regarding paper and electronic prospectuses, overall context of design, the hypertext medium, problems associated with hypertext design, and ways in which these problems might be resolved. This leads to a rationale for the research, Chapter 3, which discusses the implications of the literature review and argues that given the existing situation regarding hypertext electronic prospectus design it is worthwhile conducting an investigation of aspects of the design process and methods used in order to create a hypertext development methodology applicable to electronic prospectuses. An overview of the methodological approach is also given. The three different data gathering areas are then described, giving methodologies used, results and implications for a prospective development methodology. Chapter 4 considers the existing forms of design guidance, guidelines, methodologies and models, and goes on to describe a systematic analysis of this material. Chapter 5 examines the author's experiences with electronic prospectuses, and Chapter 6 presents a case study of a team of electronic prospectus designers. In Chapter 7, the way in which the various data were used in order to produce a provisional electronic prospectus design methodology is described, and the resulting design methodology presented. Finally, in Chapter 8, the success or otherwise of the research undertaken is evaluated, and suggestions made for future work in this area.

2: Literature Review

2.1 Introduction

The main concern of this thesis is with the design of hypertext. This chapter accordingly begins by examining the literature regarding hypertext and hypermedia in general. Issues here include the historical context, questions of definition and categorisation, with a discussion of typical hypertext features. The advantages and disadvantages which have been claimed as attributes of hypertext are considered. As the study is centred on design aspects of hypertext electronic prospectuses, it then goes on to examine motivations behind, and characteristics of, the use of new media in higher education marketing and information provision, including the use of video and the World Wide Web (WWW). This leads to an exploration of the potential and actual application of hypertext in the creation of electronic prospectuses, beginning with an account of the existing paper-based prospectus and going on to an account of efforts to produce electronic prospectuses. Attention is then turned to wider design issues with an examination of the literature regarding general design. After establishing the design context, problems in the creation of hypertext systems are considered, first generally, and then specifically with regard to hypertext electronic prospectuses. Finally, various aspects of hypertext design, including writing and print publishing, human-computer interface design and information systems design, are examined.

2.2 Hypertext and hypermedia

The storage and presentation of textual information was not initially a goal of computer function. However, this has become a major function of computing today, with a number of text handling applications and formats becoming well established, including ASCII and other text files, databases, spreadsheets and presentation software. Over the last decade, a variety of alternatives have risen to prominence, usually termed hypertext, hypermedia or multimedia, although as we shall see these have a longer history. These have been used for a variety of purposes, including electronic encyclopaedias, computer-aided learning (CAL), problem exploration systems, on-line help and documentation. Distribution can be via networks, computer disks and CD-ROMs.

2.2.1 DEFINITIONS AND HISTORY

It sometimes seems that there are as many hypertext and hypermedia definitions as there are systems. It is not this author's intention to complicate the situation further by offering new definitions, but some discussion of this area is worthwhile in the interests of clarity. The term 'hypertext' has been used at various times to refer to a wide range of systems, and the terms 'hypertext' and 'hypermedia' are at times used exclusively or interchangeably. It is also useful to clarify the distinctions which have been made between hypertext and hypermedia, and multimedia, with which they share some common features.

2.2.1.1 Hypertext definitions

As Horton (1990) says, defining hypertext and hypermedia is hard. The term 'hypertext' was first used in 1965 by Ted Nelson, although the concept was older, going back to Bush's speculations half a century ago (Bush 1945). Nelson defined hypertext as '. . . a body of written or pictorial material interconnected in such a complex way that it could not conveniently be presented or represented on paper' (Nelson 1965, in Horton 1990, p291), and also as 'a combination of natural language text with the computer's capacity for interactive branching or dynamic display . . .of a non-linear text . . . which cannot be printed conveniently on a printed page.' (Nelson 1967, in Conklin 1987 p17).

Most definitions tend to focus on either computerised links or non-sequential text as the essential feature of hypertext. Nelson's first definition emphasises the computer; the second definition highlights the non-sequential aspect, an aspect over which there is some debate, and merely implies the existence of linking in its reference to 'interactive branching'. Conklin's later definition emphasises linking: 'This article focuses on machine-supported *links* (both within and between documents) as the essential feature of hypertext systems and treats other aspects as extensions of this basic concept.' (Conklin 1987, p18).

Other definitions have depended on highlighting the non-sequential nature of hypertext, for instance Yankelovich, Meyrowitz and van Dam (1985), but as McKnight, Dillon and Richardson (1991) point out much printed text is also non-sequential in use. If hypertext is defined purely in terms of its non-sequential and linked text aspects, it is clear that a great deal of much older material could be regarded as hypertext. For instance, the traditional encyclopaedia is not intended to be read consecutively, but consists of shorter items linked to related items by means of in-text references and footnotes. To consider printed text as hypertext would be an over-extension to a far wider realm than it is really meant to. The key point is that hypertext is a purely electronic form. If one follows Wittgenstein's dictum 'don't ask for the meaning; ask for the use', one finds that use of the term supports this interpretation well. The term 'hypertext' is almost invariably used with regard to computerised systems which allow the user to follow links quickly and easily.

One way out of the definitional dilemma is to try to define hypertext in terms of a set of key features. Conklin (1987) attempts this, as do Akscyn et al. (1987). Such features include, in Conklin's example, a network of nodes, a one-to-one correspondence between nodes in the hypertext database and their on-screen representation, a windowing interface, with windows containing link icons, the capacity for the user to author new nodes and links, and three alternative ways of accessing the database: following links, searching on keywords, strings or attributes, or navigating through the material via a graphical representation of the information network. This can be summarised as a text database, a semantic net and an interface (Rada 1989). There are several problems with this feature-based approach. Firstly, what features should be included? This argument is complicated by the close relationship between hypertext and hypermedia. Should graphics be integral to hypertext, or is this properly a feature of close relative hypermedia, for instance? There also seems to be a strongly subjective element at work. Secondly, many applications which are recognisably not hypertext share hypertextual features to varying extents, for instance contemporary word processors such as Microsoft Word support document linking, keyword searching, a windowing interface, etc. Thirdly, it is noticeable that these definitions are material and technological in kind, focusing on the system rather than its content or its use. We might be surprised if on asking a colleague to define a book we were told that it was a number of paper pages sandwiched between two thicker cover pages, covered with a variety of symbols. This is true as far as it goes, but conceals much about the nature of books, the kind of information they contain, the way they feel and the uses we have for them.

For the same reasons hypertext definitions which focus on features and functions also seem inadequate, which is presumably why Nielsen, after attempting to define hypertext in terms of its non-sequential characteristics, is at one point forced to say that 'When asked whether I would view a certain system as hypertext, I would not rely so much on its specific features, command or data structures, but more on its user interface "look and feel"' (Nielsen 1990, p4). To Nielsen this is a characteristic composed somewhat imprecisely of a combination of factors such as rapid access to data, a user-friendly interface and low cognitive load in navigating through the system. This inadequacy was also noted by Knuth and Brush (1990 p93), who commented that 'We found it impossible to find a set of features (other than browsing and authoring) which were embodied by all hypertexts.' They went on to suggest that it was easier to categorise hypertext according to functional family resemblance rather than by necessary and sufficient features. Rather like Nielsen they went on to add that this may mean that a hypertext system or document is recognised, somewhat unscientifically, by 'feel' rather than any defined characteristics (Knuth and Brush 1993). This approach is unsatisfactory in that it does not make for a very tight definition, but has the advantage of avoiding arguments as to what actually constitutes 'real' hypertext. In the context of the present work the author has adopted what might be seen as a soft position, that the term hypertext can be validly applied to any system of electronic text handling in which the dominant mode of use is the activation of links for user-controlled travel between discrete chunks of textual information, regardless of universalisability or whether the reader can amend the text or see a map.

2.2.1.2 The medium and its history

2.2.1.2.1 Historical context

The beginnings of the hypertext idea can be discerned in Bush's concept of the Memex (Bush 1945), a document handling system conceived but never realised as a set of interlinked microfilm documents, but the term and the earliest systems, such as Engelbart's NLS/Augment and Nelson's Xanadu, first appeared in the early 1960s. Conklin (1987) describes these systems as 'macro literary systems'. They were grandiose in scope and intent consisting, at least hypothetically if not actually, of huge bodies of information bound together by a simple standard interface. In the 70s, attention turned to problem exploration systems, such as Rittel and Webber's IBIS, and structured browsing systems (Conklin 1987). Since then, the trend has been towards general hypertext tools which can be tailored towards a variety of purposes, or alternatively towards the adoption of hypertext facilities as ways of accessing information in a range of applications, for instance the provision of on-line help in computer software packages and as browsers in multimedia reference works. A recent innovation that in some ways corresponds to Nelson's original conception of a 'docuverse' in which all documents are linked together is the Internet-based World Wide Web (WWW), which is a structure of servers all over the world running files linked via a common mark-up language.

2.2.1.2.2 The growth of hypermedia - another problem of definition

As hypertext systems have grown in sophistication, or at least in ambition, and computer technology has advanced, so it has become possible to integrate non-textual information such as graphics, sound and video. This combination is often referred to as hypermedia. For instance, Hardman (1988) defines systems incorporating media such as video as hypermedia , and Yankelovich et al. (1985) use hypermedia to denote hypertext functionality with added graphics, spreadsheets, video, sound and animation. This presents a further definitional problem. What distinction, if any, should there be between hypertext and hypermedia? Some consider that hypertext is itself adequate. Jonassen always uses the term hypertext, even when the information fragments include graphics, sound and video, and other computer program elements (Jonassen 1991). Yankelovich et al. (1985) reserve hypertext for purely textual elements, with hypermedia covering extensions into graphics, sound and video. Nielsen (1990) argues that the two terms are effectively interchangeable, but that the 'traditional' hypertext is a perfectly adequate term in itself. In practice, he actually uses the two terms interchangeably, admitting to a preference for 'hypertext'. Separating the two terms, reserving hypertext for the more bookish text documents and systems, and applying hypermedia to those with graphics, sound, video and animation makes some sense. It is easier to identify systems which restrict themselves to text alone if the term 'hypertext' is retained for these. There is an extensive history of such systems, usually the so called 'first generation hypertexts' with an accompanying literature in which 'hypertext' is the usual term. But graphics have long been a part of the world of print, and it is not hard to extend this notion to consider graphics as part of hypertext. In practice it seems that the two terms are in use concurrently and often interchangeably. The trend so far suggests that in the long term hypermedia will become the dominant term, with the term having been adopted as an umbrella term for all types of electronically interlinked material, and in academic journal titles. As far as the present work is concerned, the term 'hypertext' has been used for the most part, but this does not imply a concern with electronic text alone.

2.2.1.3 Hypermedia and multimedia: the difference

In recent years, multimedia has become a significant area of development. This appears to have been initially technology-led, but multimedia is beginning to have an impact in industry, education and the home. It is necessary to clarify the relationship between multimedia and hypertext and hypermedia. According to McKerlie and Preece (1993), definitions of multimedia tend to involve an emphasis on the use of several media, integration or combination of these media, and interaction. Whilst hypermedia and multimedia may include the same media elements, such as video, graphics, animation and sound, the type of interaction which the user has with the computer is not necessarily the same. A multimedia presentation may allow no or limited user control. Even interactive video systems, which may allow a high degree of user control, are not necessarily hypermedia, in that they do not always produce the same feeling of total control and ability to explore the system that Nielsen (1990) finds so typical of hypertext. Here it is useful to consider hypertext and multimedia as separate domains, and to consider hypermedia as the area in which these two domains intersect. However, another perspective is to consider hypermedia as being an essential integrative device for multimedia elements.

2.2.1.4 Hypertext variety

Although the term hypertext is a useful generic term, it conceals a number of significant differences between hypertext systems and documents. Just as in paper-based texts we can distinguish between the various forms of books, including directories, text-books, novels, and encyclopaedias, and filing cabinets of documents, so we can distinguish between for instance large 'industrial strength' multi-document hypertexts for the storage of technical information and small documents for educational purposes, or between electronic prospectuses and CD-ROM encyclopaedias. Exactly what distinctions are made vary considerably. Differences occur in several dimensions: size, medium of distribution, media employed within the hypertext, software platform, degree and kind of interaction, structure, function and application, and features. Nielsen (1990) identifies a wide range of domains where hypertext has been applied, including computer applications such as on-line help, business applications, intellectual applications, educational applications, and entertainment. Conklin takes a functional approach identifying four main areas of application - macro literary systems: large on-line libraries with hypertext links; problem exploration tools for sorting out unstructured problems in writing, design and programming; browsing systems: easy to use smaller scale versions of the macro literary systems, and general hypertext technology, for a range of tasks including reading, authoring and collaborative working (Conklin 1987). Carlson (1990) suggests that hypertexts may be classified by the cognitive activities they support, and by the nature and design philosophy of the underlying software platform. In the former case, she identifies four sets of cognitive activities: reading, annotating, collaborating and learning. In the latter she suggests four categories which vary greatly in terms of their underlying architecture, but which will all look like hypertext to the end user: those which begin with static text on a word processor platform and convert it to a hypertext representation using the original text's mark-up language (SuperBook); the 'notecard' metaphor (NoteCards, HyperCard); document databases, and AI software marrying techniques such as expert systems with document databases.

An important distinction, both generally and in particular for this work, is that between hypertext systems and the artefacts produced or linked by such systems. A hypertext system is a piece of software usually, although not necessarily, intended both for reading and authoring hypertext documents, for instance Guide or HyperTies, or less commonly for linking documents made with other applications, such as Microcosm. The products of such a system are variously referred to in the literature. Hypertext or hypermedia document is common. Another option is the simple 'hypertext', as used by Wright (1990). Kahn (1989) refers to a hypermedia 'collection' reserving the term 'document' for individual items linked in the collection. 'Hypertext database' is used by Shneiderman (1989). The formulators of the HDM model (Garzotto et al. 1993, Schwabe 1993) variously use the terms 'application' - which risks confusion with such terms as 'software application', 'hypertext', or 'hyperdocument'. The author's preference is for 'document', and this use is maintained throughout this thesis. Terminology apart, a hypertext document or application could be a product of a hypertext system, or it may be a stand alone document specially constructed using a programming language such as C. A grey area exists when an existing text file produced by some other form of text editor, which would normally be considered a document in its own right, may be linked with other documents by an external piece of hypertext software like Microcosm (Hill et al. 1993). Here the document now takes on the status of an individual node within a larger hypertext document. A further point is that a hypertext document or application could be a read-only document, or one capable of editing and expansion.

Another distinction identified in the literature is that between read-only hypertext documents and those capable of, and often specifically intended for, alteration by the readers. As McKnight, Richardson and Dillon (1989) pointed out, many of the earlier hypertext systems were concerned with the production of collaborative authoring environments, and this remains an important goal for some systems. In these systems, documents are the product of work or study situations where readers also annotate, comment upon, add links and otherwise change documents. But the recent growth of the CD-ROM market and the use of hypertext for on-line help, for instance in Windows, has caused a proliferation of read-only documents.

This is significant for this work in that electronic prospectuses are most likely to be read-only documents, although developments such as the WWW may alter this position. The distinction between read-only and authorable documents reflects a crucial distinction between author and reader. Of course authors are also readers, and vice versa, but the activities undertaken in each role are very different, one being the converse of the other. The focus of this research is on the authors, rather than the readers. Using computer systems generally and more particularly reading electronic text has been studied in some depth since the advent of the related technology, although Dillon (1994) suggests that conclusions about the users of such systems and documents are often based on poor evidence and have limited value. The role of the author or designer is less well considered. Whilst programming and software design has been studied (e.g. Weinberg, 1971, Mayer 1988), and principles of software engineering evolved to aid the designer's task (Sommerville 1992), the authoring of electronic documents has had less attention. For instance, McKnight et al. (1989) suggested that there was a tendency for other writers to neglect the author's task and to concentrate on an obvious and substantial difference in reading from paper and from electronic documents, or alternatively to assume that the reader was also actively engaged in writing hypertext. However, with read-only hypertext where a document is fixed and unalterable, it is essential to consider the author's decisions, because these concern not only the information content and structure, and the interface to this material, but also the nature of the links that are inserted in the document. The reader can only use the links that are provided by the author (Raskin

1987). These not only control access to the information, they encode information. The links that an author inserts say something about the nature of the relationship between linked material. With a read-only hypertext document, static hypertext (Oren1987), it is difficult or impossible to bypass the author's links. If these links are poorly made, then the reader has no alternative but to suffer the consequences. This issue is further explored in section 2.5.3.2 below.

The point of this exploration of hypertext types is to highlight the variation that exists in hypertext in terms of systems, documents, readers and authors. This is important as it relates to methodological issues regarding the extent to which work on specific kinds of hypertext generalises to other areas.

2.2.2 ADVANTAGES AND DISADVANTAGES OF HYPERTEXT

Before considering the advantages and disadvantages of hypertext one needs to consider the basis for comparison. Such comparisons may be in terms of medium, for instance, comparing hypertext with paper or video. Alternatively, they may be in terms of area of application, for instance comparing hypertext with conventional databases, or with educational texts. Both such dimensions of comparison may be required if one is trying to gauge whether hypertext is a viable solution to a problem. Another issue is that of perspective: the reader's or the author's. What is good for the author may not be so for the reader, and vice versa. Then there is a question of appropriate and accurate evidence. So far there appears to be no conclusive evidence that hypertext is better or worse than print for instance, and it would appear that much depends on the nature of the task involved (Dillon 1994).

Many claims about hypertext are not firmly based on empirical evidence, but are still interesting. A number of advantages have been asserted for hypertext. It supposedly enables large networks of complex information to be accessed easily and intuitively, as opposed to the analytical and systematic approach required for good results from on-line databases. This encourages incidental learning and serendipity (Marchionini and Shneiderman 1988). From the reader's point of view, a good hypertext has the particular advantage that a body of information may be used in a way that is natural and easy to the user. Connected and related information is literally at the finger-tips, as opposed to scattered through an array of paper documents, or in a database field dependent upon being searched with a relevant key-word, which may or may not be apparent to the reader. There is a high level of user control, as opposed to that in a database, even in a read-only hypertext. From the author's perspective, the writing process may be supported by a range of tools for annotation, bookmarks and browsing between documents (Marchionini and Shneiderman 1988) From the publisher's point of view, the easily amended electronic form means that there is potential for savings in terms of time and money where amendments are required. There is also the potential for huge amounts of information to be squeezed into a small physical space. Hypertext also has a role in the development of multimedia, offering a highly flexible method of integrating media types, and hypertextual approaches can be seen in a number of recent multimedia CD-ROM publications.

A number of disadvantages to hypertext have been raised. Most of these issues are addressed later in the context of hypertext design problems, but there are a number of criticisms of the fundamental concept of hypertext which can be considered here.

The earliest extensive critique is that of Raskin (1987), who saw hypertext as 'one part inspiration and nine parts hyperbole' (p325). Questions he raised concerned the technology, social, legal and economic problems, problems with synonyms and difficulties in knowing what lies at the other end of a link. Speed of the system was a concern, the problem being that if a hypertext system is allowed to grow organically, the number of referents will expand to the point where the system eventually becomes unattractive to use. Links were either cumbersome, wrong or trivial, and interface concerns were only vaguely addressed.

Many of Raskin's concerns were addressed at the 'macro literary system' conceptions of hypertext which were prevalent at the time of writing, and others such as the difficulty in knowing what lies at the other end of a link are problems in representation rather than major conceptual obstacles.

Meyrovitz (1989) suggested that the relative failure of hypertext at the time he wrote was the result of a failure to find a universal paradigm corresponding to the Cut and Paste paradigm which would allow integration of hypertext with other software at work. Lack of consistency and compatibility between hypertext systems makes both reading and authoring unnecessarily difficult. This criticism was clearly a product of the kind of project - Intermedia - that Meyrovitz was working on, one which supported both reading and authoring, as opposed to read-only hypertexts, and does not raise serious conceptual concerns.

Another conceptual criticism is that of Brown, (1989b) who suggests that hypertext links have the status of 'goto' links in programming, which leads to poorly structured documents in the same way that use of 'goto' causes poorly structured programming. Brown uses this as an argument for highly structured hypertext, in particular the hierarchical structure employed in Brown's hypertext system, Guide, in which cross-hierarchical linking is hidden from the user to avoid confusion.

A more general concern, not specific to hypertext, is that there are problems associated with reading from screens, although the evidence is far from clear. It appears that problems may be experienced in five areas: speed; accuracy, fatigue, comprehension and user preferences (McKnight et al. 1989). Some experimental evidence exists which attempts to compare hypertext with other forms of information presentation. Whilst some studies purport to show no difference in reading from screens and reading from paper in certain situations, Dillon's (1994) summary of the evidence suggests that this is far from certain, as most studies have been conducted using seriously constrained tasks that do not effectively represent reading as practised. Reading studies are only of value insofar as they relate to actual tasks. Not all the evidence examines all these factors. A little evidence exists to suggest that reading from screens can be as fast as from paper where high resolution screens are used (Gould et al. 1987). There are studies which indicate user preference for hypertext, for instance Egan et al. (1989). Generally speaking it seems reasonable to state as McKnight et al. (1991, p18) do, that '. . . it is clear that for some texts and some tasks, hypertext is not the universal panacea which some have claimed it to be.' It can be argued that hypertext suffers from its unfamiliarity: whereas we have over 400 years of print experience, hypertext is only just beginning to spread into wider use. Readers and authors have no pool of experience to draw on. No test that controls for this learned experience has been applied.

From an authoring point of view, hypertexts raise certain problems. These are a major concern of this review, and are explored in section 2.5 below. In the meantime, it is useful to examine what is special about hypertext, and to identify the features that are particular to it.

2.2.3 HYPERTEXT FEATURES: LINKS, NODES STRUCTURE AND NAVIGATION

As indicated above in the discussion of definitions, certain features are normally characteristic of hypertext. A further exploration of these characteristics is useful because they are reflected in the typical design problems and issues encountered by hypertext developers. Such problems and issues are dealt with later in the chapter.

2.2.3.1 Links

Links are a distinctive, even defining, characteristic of hypertext, the means by which the associated chunks of information are connected to each other. Links and linking are consequently a focus of much work on hypertext. They are crucial to the success or failure of hypertext documents, because they reflect relationships in the body of information that the hypertext document represents. The more closely that links represent these relationships the more successful the document will be (Schwabe et al. 1992). Essentially a link consists of a directional connection between two chunks or nodes of information. Each link has a start point or anchor, and an end point. Anchors may consist of words, phrases or paragraphs, whole screens or parts of screens. Destinations may be whole screens or parts of screens such as windows. The link may be physically independent of the anchor and the destination node. The literature shows various approaches to link classification. Some of these are more formally presented than others, and have received less widespread application, for instance

Garzotto et al. (1993) distinguish three link categories within their HDM hypertext model, perspective links, structural links, and application links. A more widely accepted distinction, found for instance in Brøndmo and Davenport (1990), and Shneiderman et al. (1991) is that between uni-directional and bi-directional links. Uni-directional links are, as their name suggests, links which simply provide a link from node A to node B, without having an associated return link. Bi-directional links are those that provide an associated return link. Another useful distinction is that between explicit and implicit links (Nielsen 1995), the former being specifically authored, and the latter being available via search or otherwise computer-generated. Links may be allotted to types by the author (Halasz 1988). More extensive classifications have been developed, an interesting example being DeRose's taxonomy of links (DeRose 1989), which shows very clearly how closely link types are related to questions of structure.

2.2.3.2 Nodes

Links mean nothing without nodes, the chunks of information that make up the hypertext content. These are normally single concepts or ideas, and typically take up a single screen or window (Conklin 1987). Nodes may be frame-based, where there is a standard node size, which may be part of a screen or a complete screen, for instance HyperCard or ToolBook, or window-based, where the node can be of any size and require scrolling to access, for instance Guide. The way in which nodes are handled is important when it comes to deciding issues such as how much information should be in a node, and where topics should be split in order to fit into nodes.

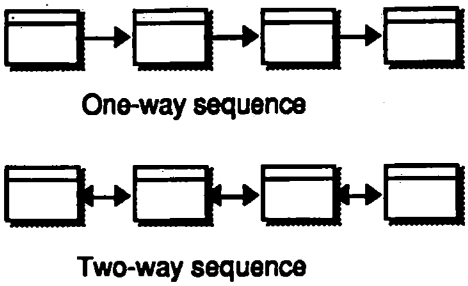
2.2.3.3 Structure

Hypertext structures can be divided up into a number of types. A number of such divisions have been proposed in the literature, using varying terminology, and of various levels of sophistication. For instance, Jonassen (1986) proposes a model with three levels of hypertext. Level 1 is chunked or node-linked hypertext, in which direct access to all other nodes is possible from each node. This is sometimes known as web hypertext, or, less flatteringly as 'spaghetti hypertext'. As the latter term suggests, this approach suffers from a tendency to cause confusion as a result of the proliferation of links in anything but a small system. It requires good support tools, such as indexes or maps, to avoid disorientation. Used correctly, however, web hypertext can supposedly give the user the opportunity to browse through material according to interest. It supports serendipity, the accidental discovery of useful information.

Jonassen's Level 2 is structured hypertext, in which blocks of nodes have limited links to each other, and are independently structured. The Level 3 hypertext has a hierarchical structure, similar to that found where menus and sub-menus are nested, or in the typical Apple operating system file structure, but with - sometimes - cross-links between nodes at the same level in the hierarchy. The problem with hierarchical hypertext is that if this structure is rigidly adhered to, there is a tendency to lose one of the more attractive features of hypertext, its ability to break out of rigid structures and link to material which is related in ways other than those of family, sub-set or super-set. Despite this, hierarchical structures are well-regarded. They have the advantage of being based on a structure of organising information which is very common in both the print medium and the world of science, and consequently can pose minimal navigational demands if implemented well. Herrstrom and Massey (1989) propose that a 'hierarchical array' is best suited to the needs of 'task-driven' users. It is capable of revealing its structure through maps, icons and similar devices, and an accurate mental model is reinforced in the user's mind with repeated use. This type of hierarchical model was adopted by the Guide hypertext system because of its ease of use and understanding (Brown 1989b).

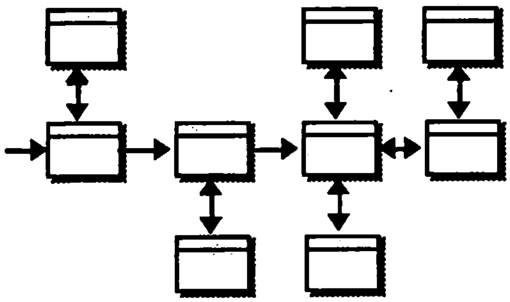
A similar model of hypertext structures is given by Brockmann et al. (1989), which relates structure to expressive power and the risk of confusion. In this model, sequential structures are low in expressive power and in risk, whilst web hypertext, similar to Jonassen's Level 1 hypertext, is high in expressive power but also in risk.

Horton (1994) gives a more extensive classification of structure types. There is the sequence, at its simplest a linear structure (Figure 2.1).

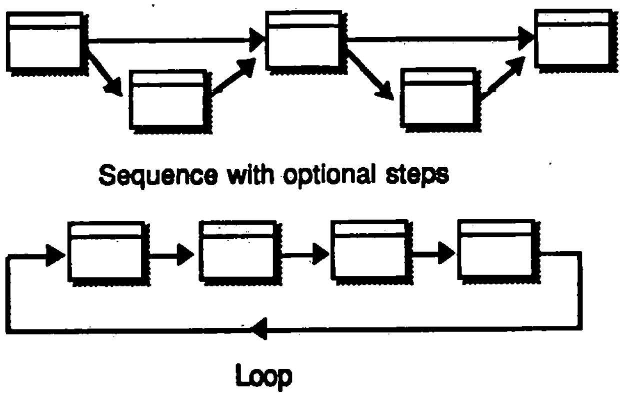


(Figure 2.1) Linear hypertext (after Horton 1994)

This can be varied in a number of ways, such as with side excursions (Figure 2.2), optional side-steps, or with a loop returning to the beginning (Figure 2.3).

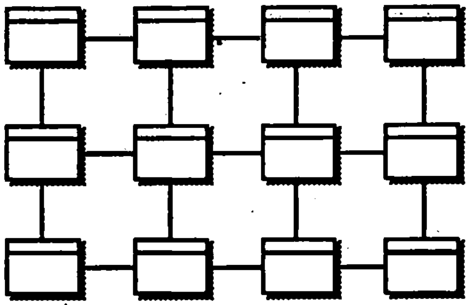


(Figure 2.2) Linear hypertext with side excursions (after Horton 1994)



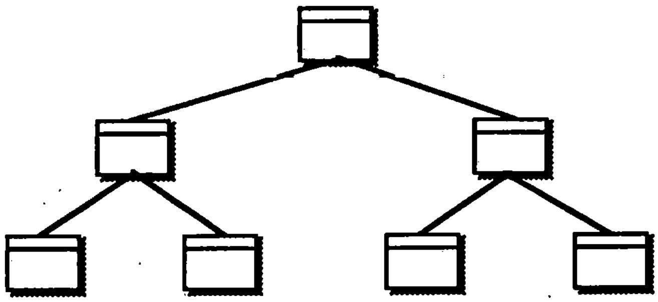
(Figure 2.3) Further linear variations (after Horton 1994)

More complex alternatives include grids or orthogonal structures and their variations. These reflect the familiar table and matrix structures used in the presentation of much printed information (Figure 2.4).

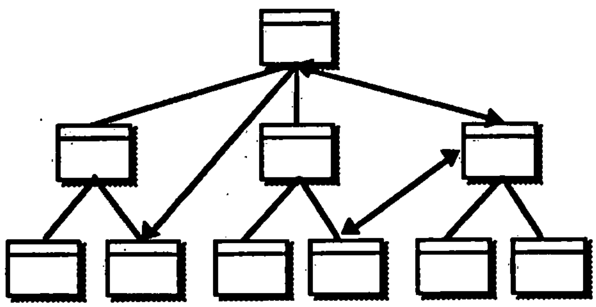


(Figure 2.4) The basic grid (after Horton 1994)

Hierarchies are classified by Horton as including a range of structures from the pure hierarchy (Figure 2.5) to lattices and hierarchies with cross references (Figure 2.6).



(Figure 2.5) The pure hierarchy (after Horton 1994)



(Figure 2.6) Hierarchy with cross-references (after Horton 1994)

Finally he distinguishes a range of web structures. In a pure web, everything is linked to everything else, a high-risk structure which ultimately renders the existence of links meaningless. Hence, like Jonassen, Horton identifies the existence of a wide range of more practical combinations of structures based on a web structure, the partial web. This allows the use of the other structure types above as sub-structures within a web structure, thus allowing a high degree of structural choice and variation without the disadvantages.

The design consequences related to nodes, links and structures are discussed in more detail below.

2.2.2.4 Hypertext models

Attempts to order and categorise the hypertext network are useful, but in concentrating on the hypertext net they do not capture all the elements of a hypertext system. As well as a net of links between nodes, a hypertext consists of the nodes themselves and their information content, and the human-computer interface. The network of links is one retrieval structure; there may well be others.

Various models of the complete hypertext system have been proposed. Frisse and Cousins (1992) summarise informal descriptions of hypertext by Conklin and others in terms of a database of text, a semantic net connecting text components, and tools for creating and manipulating the combination of text and semantic net. This level of description is adequate for many purposes. However, because of the variation in hypertext forms that exist, and the consequent difficulties of comparison and communication, and also because of a perception that formal standards may ease the adoption and wider use of hypertext a number of efforts have been made to develop models of hypertext systems. Such models are abstractions that attempt to provide a standardised hypertext terminology and structure.

Campbell and Goodman (1988) proposed that hypertext systems could be broken down into three levels, rather like the classic database architecture model:

- Presentation layer: user interface
- Hypertext Abstract Machine (HAM) level: nodes and links
- Database level: storage, shared data, and network access

This is somewhat idealised: most hypertext systems vary from this in some way, and may even be barely recognisable as hypertext in their underlying architecture. For instance, some systems (e.g. Microcosm) muddle the picture by having a database of links. In more extreme cases they are based on alternative technologies such as expert systems. It can also prove difficult in practice to separate the links from the interface, as variations in link types require some parallel representation in the interface display.

A more developed example of a hypertext model is the Dexter Reference Model (Halasz and Schwarz, 1990). As the name suggests, this was originally developed as a basis for the comparison of hypertext systems rather than applications. Again, this initially divides the hypertext system into three layers: a storage layer, a run-time layer and a within-components layer. The storage layer is a database of links and nodes, which are known as components. Aggregations of components are also considered as components in their own right. Every component has a unique identifier. Links connect nodes. They are attached to the nodes by anchors. The run-time layer is the level at which components are instantiated, that is make their appearance to the user, and at which link markers appear. So this level could be considered as the user interface level. The within-component layer is where conventions regarding the handling of media types are considered. There are some similarities with the Campbell and Goodman model, but the Dexter Reference Model adds two inter-layer elements: the Presentation Specifications and Anchoring. These treat the relationships between the different layers as being separate entities. The Dexter model was not intended to serve as a blueprint for development in itself, but a number of developers have used the Dexter model as a starting point, for instance Grønbaek and Trigg (1994) developed their DeVise collaborative design hypermedia system using the Dexter model. Other models which fulfil a similar abstracting function include Garzotto et al's HDM model. In contrast to models like the Dexter Reference Model, this was intended to act as a framework for development of applications as opposed to systems. It enables description of material in existing or proposed hypertext applications in a system-independent manner. As such it has more in common with database and systems analysis techniques for describing conceptual and logical structures, particularly the Entity-Relationship model. One point that is apparent here is that the boundary between models and methodologies is often blurred. HDM provides at least in part a systematic method for hypertext application development, so assumes some of the characteristics of a methodology.

However, in the ISDM domain, a model is seen as being an abstraction underpinning a methodology's view of reality (Avison and Fitzgerald 1988), and a model like HDM accords closely with this conception. HDM has more recently been developed into OOHDM, Object Oriented Hypertext Design Method, based on HDM, and reflecting this relationship (Schwabe and Rossi, 1995).

2.2.2.5 Navigation and browsing

Associated with questions of structure is the issue of navigation or disorientation. The question of how the user identifies location in each hyperdocument proved a major issue in earlier hypertext debate and inquiry. These issues are dealt with in more detail in the section on hypertext design problems, 2.5.3.2.4.

2.3 Hypertext electronic prospectuses

Hypertext electronic prospectuses were chosen as a focus for the study of hypertext design methodology for reasons which are explained in the next chapter. The next sections examine their context of use and the limited literature which exists on them.

2. 3.1 PAPER PROSPECTUSES

2.3.1.1 Background

The conventional paper university prospectus does not appear to have been considered worthy of much attention outside of those concerned with its production. There are few references to it in the literature except in passing in the literature relating to the marketing of higher education, and Whitby's (1992) manual for prospectus production. Yet the paper prospectus has existed in something like its contemporary form for many years. In recent decades the presentation has become more stylish and polished, a result of market pressures in the higher education (HE) sector and the increased availability of new print technologies, but the essential form and function of such publications has not changed substantially.

2.3.1.2 Function

The typical prospectus exists to satisfy the needs of the publishing institution and a varied range of users. In terms of document type, the prospectus falls into a broad category with brochures and catalogues, documents that are intended both to inform and persuade. Seen in marketing terms (Kotler 1988), the prospectus is a promotional tool informing about the product (i.e. the institution and its courses), the place (the town and region) and the price (being the requirements for entry, cost of living etc.).

The prospectus also fulfils a public relations function. Here the institution is interested in conveying the less tangible aspects of its character, such as status, ethos, and corporate identity (Balmer and Ennis 1994).

2.3.1.3 The users

For the user the prospectus is a source of information that ultimately is intended to support the prospective student and others closely involved in a decision-making process. These users include the students, close friends and family, teachers and lecturers, careers advisors in school, college and community.

Two main sources of knowledge about the potential student population exist. Several surveys of potential students' perception of higher education have been made. Keen and Higgins (1990, 1992) have surveyed young and older potential students, exploring students' perceptions of the English higher education system. There is also some focus group research with applicants (Whitby 1992). The Scottish Office Education Department (SOED) has conducted a telephone survey of potential students in Scotland (Scottish Office Education Department 1993). Other interested parties such as friends and family are less studied. These matters are examined in more detail in Chapter 5.

2.3.1.4 Structure, format and Information Content

Whitby (1992) states that the current information content of prospectuses varies widely, and lists 22 typical items. Prospectus structures typically reflect this variety. However, most prospectuses have common structural features. These are dictated to a large extent by the information content, which is broadly similar for most institutions. A typical structure consists of an introduction to the institution followed by description of the surrounding environment, services and facilities, faculty and departmental information, details of qualifications required, fees, funding and applications procedures, and course information. The amount of information may vary considerably, and its form may include substantial visual elements such as photographs, maps and diagrams. The sequence may change, with general information appearing at the beginning and end of the prospectus, or with course information at the start.

This information is typically structured in sections. A contents page(s) is usually followed by a general section including an address from the Principal or Vice-Chancellor, and the Students' Union President. This is followed by faculty, department and course details following a hierarchical pattern. There is usually a course list giving summarised information such as grades required for easy reference. Sometimes there may be an index. A good prospectus will have a logical structure, with the various contents 'sign-posted' using various rhetorical devices to locate the reader and to identify the different sections and sub-sections (Whitby 1992). There is typically some form of search mechanism, usually a contents list and index, and frequently listings by course title.

The format is open to considerable variation, with a range of presentation styles being adopted. The reasons for variation depend on the institutions concerned, but typically will reflect the institution's perception of its audience's needs, its corporate image and marketing strategy.

2.3.1.5 The designers and the design process

There does not appear to be any design advice devoted to prospectus design in the general literature devoted to design in publishing. The only significant publication on this subject is the previously mentioned Whitby (1992), which is aimed specifically at the producers of HE prospectuses. This outlines a suggested design process based on the kind of DTP design process found in the DTP literature. The designers of prospectuses and their characteristics do not appear to have been documented.

2.3.1.6 Problems associated with the paper prospectus

Where a practice has persisted for a long period with apparent success, it is worth asking what point there is in changing that practice. This is particularly the case where technological innovation is a key element. As Whitby points out, 'the written word has the advantage of being portable, accessible, substantial and visual; appropriate for both casual browsers and those seeking fuller detail.' (Whitby 1992 p iii). Given this, it is reasonable to question the need for alternatives.

Two key surveys on the English population of applicants to higher education (HE) and their relatives and friends have been undertaken. These investigated young people's knowledge of HE (Keen and Higgins 1990) and adults' knowledge of HE (Keen and Higgins 1992). They revealed considerable variation in knowledge of higher education, and in the sources used for the acquisition of that knowledge, suggesting that existing channels of communication are not entirely successful in providing information, with for instance over a third of adults saying that they had no sources of information about higher education at all. The prospectus was still by far the most important source of information for applicants still at school or sixth form college, but a much lower number of adult applicants cited prospectuses as a source.

2.3.1.7 The alternatives

A wide variety of alternative sources of information are used by applicants and those concerned with them. Keen and Higgins (1992) listed 34 sources mentioned by prospective applicants, including friends, family, existing students, visits, TV, newspapers, adverts, posters, careers conventions and careers services. From the point of view of the institution as information provider, the key alternatives to the prospectus are open days, careers conventions, videos, hypertext and multimedia distributed by computer disks and CD-ROMs, and most recently World Wide Web pages. These latter, controlled by the institution, are relevant here. All of these media have certain disadvantages. Open days and conventions require attendance in the first instance. Video, hypertext and multimedia and Web pages all require access to the appropriate technology. They also represent a considerable investment in terms of time and money. The principal alternative under investigation in this research is the hypertext/hypermedia option, and this is now considered.

2.3.2 ELECTRONIC PROSPECTUSES: A DEFINITION

For the purposes of this thesis the term 'electronic prospectus' is considered to include any on-line form of higher education course information of the kind conventionally found in a printed prospectus. The computerised form of the material could be anything that can be adapted to the purpose: databases, spreadsheets or just a collection of unlinked ASCII text files. A hypertext electronic prospectus is therefore such an on-line document containing prospectus information in which the dominant mode of use is the activation of links for user-controlled travel between discrete chunks of textual and other information.

2.3.3 HISTORY

Electronic forms of prospectus information, not necessarily hypertext, have been appearing in the US since the mid-80s, for instance at Drexel University and the Illinois Institute of Technology (Johansen 1993). Three main forms can be identified: 'electronic viewbooks', 'electronic catalogs', and electronic application forms. Electronic viewbooks typically combine graphics, photographs, music and sound, and are run on mouse-driven PCs. Some have full multimedia functions. Some of these are made using 'in-house' software, like that produced by the Illinois Institute of Technology; some use commercial products such as HyperCard. Most of these productions have been distributed until recently via floppy disk. Costs of this method of distributing information were estimated in 1993 at roughly $1 per disk (Johansen 1993). Despite this cost saving compared to video or paper prospectuses it appears that the main use of these electronic viewbooks has been to supplement the existing media, especially in technological institutions. These products vary in the extent to which they employ hypertext features. The 'electronic catalogs' for instance are databases of text files with simple search facilities. The electronic application forms are typically distributed on floppy disk at present, and appear to be simple substitutes for the conventional application form.

References to hypertext electronic prospectuses in the academic literature are still rare. Nowaczyk and Snyder (1993) report a comparison of a hypertext browser and a linear command-line system for transfers between college courses, and Johanson's paper discusses 'electronic viewbooks' as one of the ways in which electronic media may change the ways American colleges market themselves. In this country there appears to have been no discussion in the academic literature. The systems themselves were also scarce in the UK until recently. Most earlier efforts tended to be student projects: examples include HyperCard stacks from Lancaster University (1990) and Glasgow University. Sometimes these grew into fully-fledged projects which were distributed publicly, for instance the Glasgow University Visual Prospectus (1993). Only in the last few years have systems emerged which have been specifically commissioned, for instance the University of Northumbria's CD-ROM prospectus.

The recent development and widespread adoption of the WWW has enabled HE institutions to place prospectus information on-line. Whether this falls into the remit of this thesis or not depends, by definition, to the extent to which hypertext linking is used. So far, few examples use hypertext links within the prospectus, for instance the University of Glamorgan's Department of Computer Science prospectus pages do not include links in the material.

Current practice does not reflect the potential of hypertext for the presentation of prospectus information. The kind of information found in the paper prospectuses is of a kind that translates well to hypertext. Shneiderman (1989) suggested a set of criteria which documents suitable for development into hypertext should meet. These, called 'The Golden Rules of Hypertext', were that there should be a large body of information organised into numerous fragments, these fragments should relate to one another, and the user needs only a small fraction at any one time. By these rules, a prospectus is a suitable document for conversion to hypertext. It consists of relatively small chunks of information, and use of a paper prospectus reveals a need to cross-reference regularly between related sections in the prospectus. A further reason for considering the use of hypertext is that the material needs frequent up-dating.

A related area providing further working and documented examples is that of campus information systems. An early example of this was the Drexel Disk, which provided introductory information about the Drexel University campus for new students (Hewett 1987). However, although much of the information content may be shared with electronic prospectuses, there are differences in terms of both information content, mode of distribution and intended users. Campus information is typically distributed on-line, not necessarily the case with an electronic prospectus. The user community is different, consisting of existing students and staff plus visitors to the campus, and their information needs substantially different. The information content is - or should be - primarily aimed towards this group.

2.3.4 CURRENT POSITION

Currently available electronic prospectuses include those produced by Glasgow Computer Science Department, the University of Glamorgan, and the University of Northumbria. These are all distributed on CD-ROM disks. Many HE institutions have their own WWW pages. These are not usually hypertext in any true sense, being mostly text and graphics files with few links between them. Some, for instance the University of Bradford's are more ambitiously hypertextual.

2.4 Design: the broader context

The specific concern of this thesis is with the design and authoring of hypertext. This represents a particular and special case within the wider context of design. Hypertext design can be placed initially within the context of the design of interactive systems. This in itself is an activity within the broader context of design. Hypertext design is also an activity combining skills practised within certain specific design disciplines such as document design and graphic design. This section first examines the broader context of design as a discipline, concentrating initially on the design process and then on what designers do. It then concentrates on the nature of hypertext design problems, and examines the design disciplines which hypertext design may draw upon. It then goes on to examine some possible solutions to the design problems encountered.

2.4.1 DESIGN IN GENERAL: PROBLEMS AND SOLUTIONS

The design of traditional paper-based documents and hypertext document design are special cases within the whole design continuum. What is design? There is a wide variation in the different definitions offered, which usually attempt to define design in terms of process, and opt for a definition in terms of results. For the present purpose it is useful to adopt an approach which does not make such inherent assumptions about design, and it is probably sufficient to use Jones' quasi-definition, ". . the effect of designing is to *initiate change in man-made things*" (Jones 1980 p4) (author's italics). More broadly, design may be seen as a process of innovation (Sauer 1993). Design is a difficult activity however defined and whatever its purpose. The difficulties stem from two opposing tensions: the drive to systematise and generalise, and the realities of practice and context. Twentieth century design until recently has been characterised by the drive to systematise and generalise, culminating in the design methods movement of the post-war period, which attempted to apply various interpretations of the scientific process to design, typically adopting the methods of systems theory. The rationale was that general principles could be elucidated from practice in the various design disciplines, and then applied systematically. This usually means breaking design down into stages and more or less standardised procedures. The typical design stages found in this approach to design are drawn from the scientific paradigm, and represent a parallel to the inductive process of science: analysis, synthesis, evaluation, the rationalist tradition of thought. In this view of design, as described by Duffy et al. (1992a), analysis is the activity of characterising and understanding the design problem and situation, and planning sequences of action. Synthesis is the activity of actually building a product, and evaluation is the activity of choosing amongst a set of design alternatives, including articulating the arguments for and against, and also determining the effectiveness of whichever implementation is chosen The designer's objectives within this approach are usually seen in terms of goals or problems. Typical representatives of this systems approach to design in general are J. Christopher Jones and Herbert Simon (1981).

Jones (1970, 1980) made an extensive exploration of the design process, examining such aspects as the phenomenon of design and traditional and modern design methods, and described, evaluated and codified a number of design methods. The debate has moved on since Jones first published, but many of the points raised are still relevant and useful in understanding design.

The essence of Jones' approach to design is that design is a learning process. From this perspective, a design method is a way "to get from the ignorance-of -the-new from which one starts to the knowledge-of -the-new with which one ends (knowledge of what the problem really is, as well as solutions)." (Jones 1980 p xix). In Jones' terms, design methods are *any* actions which may be taken during design.

Earlier approaches to design were craft-based. They involved an evolutionary approach in which artefacts were gradually developed and improved. This was superseded by a drawing-based approach in which ideas were outlined and then specified using drawing as the medium. This approach dominated design in the 19th century. For a number of reasons, this approach has proved inadequate in the 20th century. Why should this be the case?

According to Jones, design is inherently problematic. "The fundamental problem is that designers are obliged to use current information to predict a future state that will not come about unless their predictions are correct." (Jones 1980 p. 9). In other words, design consists of tracking back from an ideal or desired future state to the beginning of the process that will bring that state about. In the process of carrying out the steps required to get to the desired future state, improved objectives and unexpected obstacles continually return the designer to the beginning of the process. The process itself contains strong interdependencies between different stages in the designed product's life history, which means that the designer's role is to attempt to continually avoid incompatibilities between stages by revising incompatible aims into others that are.

In his exploration of the need for new design methods, i.e. not craft- or drawing-based methods, Jones identified 4 critical questions:

1. How do traditional designers cope with complexity?

2. In what ways are modern design problems more complicated than traditional ones?

3. What are the interpersonal obstacles to solving modern design problems?

4. Why are the new kinds of complexity outside the scope of the traditional design process?

(Jones, p28)

The answers that Jones gives, here very briefly summarised, are that:

1. Complexity is traditionally dealt with by only working on one conception of the whole design at a time. This conception is examined for acceptability, and then if necessary revised. In other words, a proposed design is used as a technique for investigating the problem situation.

2. Modern design problems are more complex because they involve a broader number of concerns than was traditionally the case. Whereas designers were once concerned with components and products, they now have to deal with higher level concerns, at the level of the system, i.e. the relationships with other products, and with the community level, the political and social level. The implication of this is an increased complexity accompanied by a decrease in stability, in which the key limitations on design come from the individual in the shape of ideas, values, opinions and beliefs, and in which politics plays a crucial part.

In this situation, the role of design is no longer one of satisfying existing needs, it is one of designing for what is possible. This manifests itself at the practical design level in two forms of complexity: complexities external to the product, and complexities intrinsic to the product

External complexities include such factors as technology transfer, the prediction of side effects, the negotiation of national and international standards, and 'sensitivity to human overlap', that is the conflicts caused by the interaction of separately designed artefacts. Internal complexities include high investment costs, complex and undocumented compatibilities in organisations, and the problem of making rational design decisions in the face of rapid and massive changes in materials, technologies, needs and ideas.

3. At an interpersonal level there is a resistance to the kinds of radical design solutions that are necessary to solve major contemporary design problems.

4. The traditional design process fails because the tools formerly used for visualising the design problem are inadequate in the face of complexity, and the scattering of expertise amongst larger numbers of people with vested interests in maintaining the status quo.

In this context, design methods are central. In Jones' view of design, the design methods are the components of a course, an education, which the designer undertakes in order to complete a design. This course, the design process, is itself the subject of design. The rationale behind this is that the resulting confidence in the design process is an essential prerequisite to creativity.

In this view of design, finding methods for exploring the design situation, the search space in which we have to look for design solutions, is of central importance. This led Jones to present a number of design methods, some his own and some culled from the literature, for exploring the design situation, searching for ideas, exploring the problem structure and evaluating possible solutions.

The corollary of complexity is uncertainty. Design outcomes are not fully known until later in the design process.

Simon (1981) takes a similar stance, his project being to create a 'science of the artificial' as opposed to natural science, a design science. In many cases the goal of such a systems-led approach is to formalise the processes observed in the human situation requiring change. This explains the attractions of systems-led design for the designers of interactive systems: this easily leads to formalisation in the shape of algorithms, which makes the process of converting situation to code much easier, even enabling the use of software tools to automate the design process.

2.4.2 THE SYSTEMS APPROACH IN MORE DETAIL

Checkland (1981) suggests there are two kinds of systems approaches: hard systems thinking and soft systems thinking. Dahlbom and Mathiassen (1993) extend and modify this view to identify three kinds of systems approaches to design. The third, dialectical systems, is a highly politicised approach based on the recognition of conflict and instability in the world. Our main concern at present is with the first two types of thinking.

2.4.2.1 Hard systems

The hard systems approach involves describing a system in terms of a hierarchically structured set of elements. A system can be decomposed to its smallest significant elements, and everything above that level in the hierarchy can be expressed in terms of those elements. This approach aims to deal with the complexity of modern design problems by using functional analysis. Any system is regarded as a machine with a determinate function. Functional analysis means effectively dismantling the machine to see how it works. Even the most complex system can be understood by decomposing it into sub-systems until the smallest functional elements are found. By concentrating on function, we keep complexity to a minimum. The simplest component parts of systems are relatively easy to design, or may already exist. Design then becomes a matter of assembly of components. This kind of approach is employed in many areas of design, and is particularly common in the area of software systems development, a key example being the use of structured methodologies in systems design, for instance Gane and Sarson (1979), De Marco (1979), and Jackson (1983).

2.4.2.2 Soft systems

An alternative perspective is that of soft systems. This approach is based on the idea that in many ways our world is socially constructed. Our view of the world out there is shaped by our own beliefs and experiences. Hence our perception of systems is also shaped by our beliefs and experiences. This applies to all those concerned with a system. Those involved with a system all bring their own beliefs and experiences to the situation. Consequently the design problem becomes not one of decomposing the system 'out there' into its functional components, finding out how it 'really' works, but one of understanding and reconciling a number of varying and often conflicting views of the system. So the method associated with soft systems is interpretation and the creation of alternative systems representing the different views of the various 'actors' involved in the system, the different members of the project team, the various users, the other people with a stake in the system from those commissioning the system to those charged with its maintenance. The soft systems approach is very much in line with Jones' design philosophy, that design is a learning process. In practising the soft systems approach, the designer engages in a process of learning about an organisational situation that leads to an understanding of a number of different perspectives on a situation and finally to a point at which a number of alternative systems can be formulated. These can then be evaluated, and useful and acceptable systems implemented. The classic statement of this approach is Checkland's Soft Systems Methodology (1981).

2.4.2.3 Problems with the systems approach

Both hard and soft systems thinking can be considered as being based on a rationalist world-view in which the participants in the design activity are engaged in a rational and broadly co-operative process. Hard systems thinking is particularly rooted in the mechanistic/rationalist tradition of science, in which stability is dominant and reasonable prediction thereby possible. It suffers from particular limitations in this respect. It has therefore been criticised for a failure to deal with subjective and contextual conditions (Winograd and Flores 1986). Soft systems thinking possesses elements of the rationalist approach, but attempts to transcend the limitations of hard systems thinking with its appeal to the idea of multiple perspectives. This approach can seem over-optimistic, as it depends on a view of the actors in organisations as being rational beings with ultimately shared aims who can be trusted to become engaged in a constructive dialogue to bring about a solution acceptable to all involved (Dahlbom and Mathiassen 1993).

2.4.2.4 What designers actually do: empirical approaches to design

The discussion so far concerns what designers ought to do. What do they actually do, and how effective is this practice? What do designers feel is lacking in their practice? Design practice has been studied empirically in a number of design disciplines, particularly architecture. This shows a picture recognisably like the systems model of the design process, but more complex and with some variation in the roles of the three stages. For instance, Lawson (1980), in a series of experiments comparing final year science and architecture students, found marked differences in their problem-solving approaches. The scientists tried to get as much information about the design situation as possible in as short a time as possible, whilst the architects tried solutions early, and modified them until a workable solution was achieved. In other words, the scientists were looking for explanatory rules whilst the architects were driven by the search for a solution. Lawson goes on to suggest that whereas 'traditional' systematic accounts emphasise the linear move from analysis to synthesis, a more realistic description of the design process is one that integrates the processes of analysis and synthesis, in other words, understanding the design problem or situation comes through trying out potential solutions. Darke (1979, in Lawson 1980) interviewed architects engaged in local authority housing projects, and found that in the face of complex design situations, they tended to make an initial assumption, what Darke called a 'primary generator', such as starting with a terrace of houses, using this to narrow down the available choices and design a scheme, and then analysing the implications of the scheme. Once again, analysis and synthesis merge, or more specifically synthesis feeds back into the analysis stage.

2.4.2.6 What interactive systems designers actually do

In the area of interactive systems, studies initially concentrated on the working practices of software designers and programmers. Earlier studies tended to suggest that understanding the design problem was a separate task completed before the design process itself commenced (Adelson and Soloway 1985, Kant and Newell 1984), that is, the conventional analysis-synthesis steps were followed. However, Guindon has conducted a number of studies (Guindon et al., 1987, 1988, 1989), which broadly corroborate what Lawson (1980) and Darke (in Lawson 1980) found with architects. It should be noted that most of this research tends to involve very small design tasks which are unlikely to reflect the complexity and uncertainty of real world tasks. More recently attention has broadened to include designers of on-line help. Duffy et al. (1992) studied the design methods of 20 designers of on-line help, and found that in most cases the design process involved iterative cycles of synthesis and evaluation, with analysis included as part of this process in some cases. Most developers used some form of prototyping, ranging from just one paper-based prototype to 5 on-line versions. Most were constrained by their working situation in some way, either not being in contact with users, or having no input into the design of the user interface.

2.4.2.7 Design and science

Dillon (1994) suggests that using such evidence as Lawson's to claim that designers and scientists have different problem-solving approaches does not take into account the different aims of the two groups. Because design work emphasises solutions, designers produce solutions. Because scientific work emphasises reasons, scientists produce fewer closely reasoned solutions. Instead he discerns close similarities between the activities of designers and scientists, supporting this argument by recalling Popper's model of science (Popper, 1986, in Dillon 94). This argument is similar to Carroll et al.'s (1991) concept of the task-artefact cycle, which argues that design is in itself a research activity, with the designed artefact being, within the design context, the equivalent of a theory in science. The designed object represents the existing state of the designer's knowledge and constitutes the equivalent of Popper's conjecture.

2.4.2.8 Design and psychological theory

Attempts at psychological theories of design are limited. Lawson considers design as an example of what Wertheimer called 'productive thought' (Wertheimer 1959 in Lawson 1980 p103), and goes on to note the recurring theme of a split between rational and logical thought processes which for instance sees productive thought as dividing into convergent and divergent thought processes. Convergent thought involves deductive and interpolative skills, and is rational and logical. Divergent thought involves the open ending seeking of alternatives in the absence of a clear-cut answer, and is creative and imaginative. Wærn (1989) argues that cognitive psychology can aid understanding of a number of design tasks, including information retrieval and the assessment of relevance, integration of information, creation of alternatives, making inferences and predictions, and choosing amongst alternatives. Psychological theory as applied to interactive design is problematic, Landauer (1991) for instance suggesting that there are simply too many variables for it to be possible to have psychological theory with predictive power in an interactive design situation. Carroll et al.'s task-artefact cycle (Carroll et al. 1991) is an attempt to avoid the problems of theory in interactive design, based on the premise that technological innovation is rarely a product of deduction from scientific principles, but rather comes from a combination of emulation, evolution of detail and iteration, and that this process can be seen at work in the design of interactive systems.

What does all this mean for the hypertext designer? Before answering this question, it is necessary to return to hypertext, and to examine the hypertext design process in more detail.

2.5 The hypertext design process

2.5.1 TO WHAT EXTENT ARE HYPERTEXT DESIGN PROBLEMS HYPERTEXT-SPECIFIC?

As hypertext is a sub-set of interactive systems generally, many of its design problems and issues are common to such systems. However, it is arguable that there are a set of hypertext-specific problems and issues, and that these pertain to hypertext-specific features such as links, nodes and the hypertext net or graph. Much depends on one's stance regarding hypertext and user interface design. In a sense hypertext is a specific kind of interface, enabling particular kinds of interactions such as browsing with a body of data. This is why it is possible to consider, as Carlson (1990) does, hypertext as a front-end for a variety of different bodies of information such as databases or expert systems. If hypertext is merely an interface, then its design problems are to a large extent user interface design problems. But although this position has its logical appeal, it does not tell the whole story. There is a sense in which it is difficult to unpack the interface from the information content. As noted before, a link is not only a way of finding a piece of information, like a keyword search or an index, it also has the capacity to encode meaning about that information. This may be merely by making a connection to the information, or in more sophisticated systems, the kind of link may make certain statements about the relationships between the starting point and the linked information. At a higher level, the hypertext structure is also likely, if the design is good, to reflect the structure of the information. In other words, a hypertext system is not only an interface, and hypertext design is not merely about user interface design but about the design of the information content as well. As Waterworth says, "In a sense, hypermedia interfaces place users directly within the informational structure they are creating or exploring." (Waterworth 1993, p453).

2.5.2 WHAT GENERALISATIONS CAN BE MADE ABOUT HYPERTEXT DESIGN?

If it is accepted that hypertext design poses its own particular problems, what can validly be said about these problems, and about hypertext design in general? It is not appropriate, given the variations possible in hypertext documents, to claim that hypertext design problems are common to all hypertext documents or that there is a common pattern of design for all hypertext documents. However, some cautious generalisations can be made about these issues, and some normative statements proposed, with the caveats that each hypertext system or document is different, and that much also depends on the user's task (Wright 1990).

The literature already carries generalisations about design in various forms. Many aspects of hypertext design have been described empirically in the literature, usually in the form of reports chronicling the progress of individual projects. There is a tendency for these to be rather short on details about the actual authoring of documents and systems but rather to emphasise software features, problems encountered and evaluation results, for instance Yankelovich et al. (1985) and Egan et al. (1989). This especially applies to older material where the focus tends to be on hypertext software rather than authoring and authors are more concerned with the essential nature and actual and potential use of hypertext. Many accounts concern systems that have been built for research purposes, such as experiments with users, (e.g. Wright and Lickorish, 1990), and the focus is, reasonably enough, on these rather than on the design process. Some do give a real flavour of the problems and decisions involved in hypertext authoring, a good, if ageing, example being Brockmann et al. (1989), but even in cases where there is considerable detail about the actual project the emphasis may be on system, technical and conceptual aspects rather than on the design process (e.g. Carlson 1990). There appears to have been little attempt to examine what hypertext designers as a population do, the closest work to this being Knuth and Brush's Hypertext '89 (1990) design survey, which concentrated on the relative importance of a range of hypertext system features to hypertext designers, and Nicol's (1988) study of HyperCard stack designers, which showed a tendency for such designers to work from the bottom up rather than to work from the top down as advocated by system-oriented design approaches. Duffy et al's (1992) survey of on-line help designers concerns a group which one might reasonably expect to have some common characteristics with hypertext designers. Nor have hypertext designers been studied on an individual basis. Most reports of projects are self-reported and there appear to be no detailed case studies of design teams at work specifically on hypertext.

The literature also includes a body of normative advice, ranging from comments appended to empirical reports to more focused collections of guidelines, principles, models and methodologies. These latter are a central focus of this research, and are therefore examined in detail in Chapter 4. They include adaptations of HCI guidelines for hypertext interface design (Hardman and Sharratt 1990), principles and guidelines for hypertext in general, for instance Shneiderman's (1989) 'Golden Rules' and other 'considerations' for hypertext authoring , guidelines for specific systems (Apple: HyperCard Stack Design Guidelines, 1989), models for hypertext system authoring such as HDM (Schwabe et al. 1992), and proposals for hypertext methodologies (Perlman 1989, Van Vliet and Wilson 1993, Isakovitz et al. 1995). There are also a number of general books on the subject such as Shneiderman and Kearsley (1989), Nielsen (1990, 1995), Martin (1990), Waterworth (1993) and Berk and Devlin (1991), which contain sections of design advice and guidance, some of which include stepped or staged accounts of the hypertext design process. None of these books, whilst containing much useful material, really gives detailed procedures to guide the hypertext designer. However, together the body of hypertext material suggests that the hypertext field is maturing to a point at which some tentative generalisations can be made, but there is still considerable variation. An examination of accounts of the overall hypertext design process reveals this. It also reveals that hypertext design shares certain common elements with other accounts of design.

For instance, Martin (1990) and Shneiderman (1989) both include accounts of the hypertext design process. Martin scarcely mentions the design and authoring component of the process, choosing to concentrate on the research and materials gathering work such as obtaining resources, determining media and standards, and what subject matter should be included, whilst Shneiderman presents a more balanced process which places more emphasis on the later design and authoring aspects. Both these accounts only hint at many of the tasks involved in hypertext authoring. Martin gives a better indication of the possible extent of preparatory work, whilst Shneiderman suggests that an element of iterative prototyping work may be necessary in the form of a test phase. Martin also emphasises the importance of 'reviewing and field testing'. Both gloss over the difficulties that may be involved in structuring the information and producing an appropriate link structure for that information, and neither consider user interface issues within the context of overall design. This tendency to emphasise certain aspects at the expense of others is typical of accounts of hypertext development, and probably reflects both the difficulties inherent in making generalisations about hypertext development, and the way in which authors' disciplinary orientations and practical experiences may affect such accounts.

2.5.3 WHY HYPERTEXT DESIGN IS DIFFICULT: A PROBLEM-ORIENTED EXPLORATION

What follows is a problem-oriented examination of hypertext design. This examination is intended to show those features and problems associated with hypertext design which arguably make it worthy of study in its own right, with a body of knowledge specific to hypertext design. Problems with design in general can, as we have seen, occur for a variety of reasons. Jones (1980) has indicated a number of factors making design in the late twentieth century problematic in terms of complexity. Interactive systems in general present great complexity (Landauer 1991). More recent work indicates the complex cognitive tasks which may be involved in designing for interactive systems, involving several levels at which the design process works, and a number of stages (Waern 1989). Hypertext design manifests aspects of this complexity: it concerns the operation of computer systems, which in themselves involve a considerable number of variables, it concerns the interaction of users with these systems, it concerns the information content and structure, and it concerns the social context in which all these variables interact. As hypertext design is also a process of innovation, there is also an inevitable degree of uncertainty (Sauer 1993). Nanard and Nanard (1995) suggest on the basis of informal observation that the hypertext design process works simultaneously as a top-down and bottom-up process, a process suggested as a working method by Horton (1994).

2.5.3.1 Perspectives: artefacts and processes

These factors, complexity and uncertainty, undoubtedly have an effect on hypertext design, but the hypertext designer is likely to be concerned with more mundane and concrete problems. It is, initially at least, useful to consider the difficulties by borrowing a structuring device from Harris and Hosier's taxonomy of on-line help (1991) and consider them from two perspectives, firstly the perspective of the artefact and its users, and secondly the perspective of the information developer or author of the artefact. The first perspective concerns the usability and functionality of the artefact: whether it fulfils its purpose effectively and whether it does so in a way which is acceptable to the user. The second perspective concerns the experience of the author/designer in managing the various design processes and tasks. Whilst the focus of this review is on authoring and design, it is undesirable to forget the artefact/user perspective because the outcome in terms of the artefact and its usability is to a considerable extent a reflection of the design process used, and the kind of functionality required critically affects the design process. Conversely it is clear that the design process is affected by what is designed. The two perspectives are exactly that, two different ways of viewing the same phenomena, and too distinct a separation is not sustainable.

2.5.3.2 Problems from the artefact perspective

Looking at problems in designing hypertext the artefact, these can be structured in terms of the total concept of hypertext, existing hardware, software systems or existing documents. The evidential basis from which criticisms have been made is variable, as is the extent to which problems are amenable to design solutions.

2.5.3.2.1 Hardware

Hypertext is often criticised for hardware reasons. For instance Nielsen and Lyngbæk (1990) complained that hypertext systems running on the current generation of computers were cumbersome, fragile and consequently less convenient than the printed book for instance. Other problems relate to the handling of other media, for instance colour handling and video resolution and smoothness, and the limitations of the typically available small monitors, which hamper overviews of the complete hypertext graph. Such problems are not amenable to software design solutions in the direct sense, but hypertext design has to take into account the hardware limitations currently applicable.

Hardware poses another problem, identified by Glushko (1992a): the limitations of the installed user base. Research-based hypertext projects often employ state of the art technology that bears no resemblance to the kind of hardware and software base that is likely to be available to working organisations. The developer must work to the limitations imposed by the operational situation, even if this means that all potential functions cannot be implemented. It is also vital to design to take advantage of new technology as it arrives. This ideally means separating the information storage and the user interface of the hypertext system (and possibly the links) to enable improvement in the interface without having to dump the stored information and start again.

2.5.3.2.2 *Software*

Software systems for authoring hypertext have been reviewed and critiqued by a number of authors both individually and comparatively, although comparative studies are still a rarity. The software's effect on authoring cannot be underestimated. Kahn (1989) made a comparative study of four systems, HyperCard, Guide, KMS and Intermedia, and showed that a variety of system features could have significant impact on documents produced with the various systems. For instance, HyperCard has only limited support for many functions. Word-processing facilities are limited. Only one node is visible at a time. This means that following a link automatically replaces the starting node with a new node. The result is a tendency to linear and hierarchic structures. Other problems with HyperCard include limited and awkward colour provision and no way of keeping track of uncompleted (dangling) links. To obtain more than very limited functionality, it is necessary to use HyperCard's scripting language. To do this in any but a highly localised manner requires some programming skill. To gain many features and functions it is necessary to call routines written in other programs.

These kinds of problem are typical. McKnight et al. (1989) identified the three most common problems in hypertext authoring as links to nodes that have not been created, nodes which are not linked at all, and consequently inaccessible, and nodes which are accidentally linked to themselves. All of these are avoidable if the software offers sufficient support. There is no perfect application for hypertext. The general tendency is for hypertext systems to provide very limited tools for editing the various media. So there will be very basic word-processing facilities, limited drawing tools, rudimentary sound and video editing software. This means that these media will need to be edited in other programs, and then imported. There is every chance that the hypertext system software will not work easily with the other software providing these facilities, requiring care with file formats and necessitating any re-editing being done in the original program. This makes conversion of material from existing documents difficult (Glushko, 1992). Many systems are still only available on one platform, and cross-platform compatibility may be difficult or impossible. For instance HyperCard is a Mac-based system, whilst ToolBook is a similar PC system. Conversion is possible in one direction only, takes considerable time, and may be less than perfect. It seems that at present there is no ideal hypertext software solution, but a set of choices with various associated trade-offs. In some cases, developers choose not to use proprietary software but to write their own using a programming language or languages and other software, for instance Intermedia used a combination of a modified version of C with modified versions of the Mac ToolBox and MacApp, along with a database implemented in Ingres (Meyrowitz 1986). Much depends on the functionality required and on the resources available to the developer.

It could again be argued that like the hardware, software is not directly amenable to design solutions, although the question of which software to choose is clearly integral to the hypertext design process. However, hypertext design is to a large extent about coping with the vagaries of the software. For instance, the problems of incomplete and wrong links can be dealt with by careful design and documentation of links, including type, anchor and destination node, the rationale behind the link, and whether the link has been fully implemented.

2.5.3.2.3 *Document problems*

At the document level, a number of problems have been noted. Typical of criticisms of hypertext documents are those made by Brown (1990). Brown felt that over 50% of hypertext documents published were unsuitable for use. This was based partly on general experience and partly on reviews of student work, but he asserted that the typical faults of the student work were reflected more widely in hypertext documents. Some of the faults identified included over-use of the technology, where clever effects were employed without cause, poor visual design, a failure to use the medium to the full, lack of a coherent overall structure, poor writing, inconsistent and haphazard functions, and lack of consideration of maintenance issues. Brown has also likened the poor standard of authorship to the state of programming in the fifties, before the introduction of structured programming.

2.5.3.2.4 *Links, navigation and the user interface*

User interface design is the design manifestation of human-computer interaction, itself a multi-disciplinary subject drawing upon such disciplines as computer science, sociology, psychology, and anthropology. Its central concern is to ensure that interaction between users and computer systems is as satisfactory as possible, satisfaction being a function of safety, efficiency, usability and usefulness. The user interface is a key part of any interactive system, allowing communication between user and machine in order to accomplish a task or tasks. This presents both physical aspects, the keyboard, mouse and display, and representational aspects, the icons and metaphors utilised (Dillon 1990). Typical concerns for user interface design involve both hardware and software. They include the selection of appropriate input and output devices, dialogue and interaction styles, user and task characteristics, and environmental, social and organisational factors affecting the task context.

Quite how one relates these concerns to hypertext is problematic. The user interface is identified in models of hypertext architecture as one of the major components of a hypertext system. But hypertext and user interface design have an interesting relationship. Users have considerable control over the hypertext system and are in very direct contact with the information held in the hypermedia system. There is a high level of user interaction with the system (Waterworth 1993). There is a case for saying that hypertext is in itself a form of user interface; hence it makes sense to talk about a hypertext user interface to a database or an expert system.

This being the case, what user interface concerns are relevant to hypertext design? One can expect the usual concerns of user interface design to be involved, an emphasis on the centrality of the user, the task, and the context of use. Dillon and

McKnight (1995) have suggested that it is sufficient to apply good user-centred design principles to the design of hypertext, and that there is no need for special theory or methodologies, but hypertext arguably presents specific problems.

Firstly, these reside in what is unique about hypertext: its linking mechanisms and consequent structures. The use of linking imposes a particular hypertext-specific interaction style, involving the navigation and browsing of an information space. The nature of links and the form of the linked hypertext structure is highly dependent on the structure of the information in the document. It has been suggested that ease of use and the success which users have with a hypertext document is in turn dependent upon the structure of the hypertext network (Perlman 1989, Shneiderman 1989, Conklin 1987, Yankelovitch et al. 1985).

Secondly, as hypertext documents emphasise user involvement and interaction, the final context of users and the use to which hypertext documents are put is critical. Nielsen's study of benchmark measures for hypertext usability suggested that the two most important issues for usability in hypertext were individual differences amongst users and the effect of different tasks (Nielsen 1989a).

Links, navigation and browsing

Turning to linking first, the hypertext user is largely dependent on the links that the author (or an automatic routine) inserts. At its worst this may lead to the scenario that Raskin (1987) envisages, in which the unfortunate hypertext user is forced to 'second guess the link builder's frame of mind' (p329) and ultimately devote a disproportionate amount of time to knowing the system.

Other concerns for linking include such rhetorical issues as how to represent the existence of a link, how to represent different kinds of links, and how to inform the user what is at the other end of a link (Landow 1989). More fundamental are the questions of where links should exist, and how many links should exist. These are again closely linked to questions of structure.

The existence of linking implies a particular style of interaction, navigating through an information space (Dillon 1994). One particular aspect of using hypertext that received a lot of attention in the 1980s and requires examining is the problem of getting lost in the hypertext information space (Elm and Woods 1985, Akscyn et al. 1987). Conklin (1987) identifies the problem as one of disorientation. Empirical evidence of the problem was provided by Nielsen and Lyngbæk (1990), although this was only a small sample based on only one hypertext application. The problem is said to appear because hypermedia offers no spatial or tactile form of reference, for instance there is no physical sense of how far through the material you are as there is with a book. Consequently users were thought to become disorientated, not knowing where they were, where they'd been, where they were going. A number of tools have been developed for dealing with this problem. For instance, maps and graphical browsers, structured cues, including footprints, contents lists, colour coding and typographic devices have all been utilised. These have been investigated for effectiveness. Simpson and McKnight (1990) studied the relative effectiveness of structured cues, and found varying degrees of effectiveness. Wright and Lickorish (1990) found that different navigation systems appeared appropriate in different circumstances. They suggest that much depends on the structure represented in the hypertext, and the task which the users are trying to accomplish. Other practical problems may exist. For instance, it is very difficult to use a small VDU screen to map the contents of a large and complex document but it is possible to design hypertext systems using a number of techniques to minimise the problem, for instance fish-eye views (Furnas 1986).

Related to the navigation problem is the issue of browsing. The navigation problem varies to some extent according to the kind of activity the user is involved in: searching or browsing. The more motivated activity of searching is less subject to wandering into strange territory. Browsing refers to non-directed wandering around the hypertext system, a kind of electronic parallel to browsing the library shelves looking for anything that seems interesting or that catches the eye. First of all, should a hypertext system support browsing? It could be a very good way of getting lost. Browsing is also very time consuming. As Herrstrom and Massey (1989) put it, 'task-driven users browse on weekends. . . '. A similar point is made by Jaynes (1989). If your purpose in using a hypertext is more directed and motivated than this, browsing could be a bad idea. Yet browsing is one of the things that hypertext supports so well, and it is a way of escaping the strait-jacket of conventional information retrieval: not finding relevant material because you searched on the wrong keyword, or could not even think of an appropriate keyword. The key here is the phrase 'task-driven'. Much depends on the users' tasks and motivations. For instance there is no reason why a hypertext system should not support keyword searching or other techniques if the users' tasks require quick direct access to certain information at certain times (Wright, in Streitz 1990). More recent commentators suggest that the navigation problem has been overstated and is not intrinsic to the hypertext medium. Bernstein (1991) for instance suggests that the navigation problem rests upon flimsy conceptual and anecdotal evidence, and that many reported navigational difficulties may be a product of poor document design or complex and obscure subject matter. The navigation debate has certainly receded into the background, giving rise to the suspicion that many of the problems stemmed from unfamiliarity with the medium of electronic text in general and hypertext in particular. However, the possibility that navigational difficulties may be a product of poor design and production still exists.

Reading from screens

This has already been mentioned (section 2.3.1.4 above). It is not a hypertext-specific problem, but one which arises wherever text is represented on CRT screens, and hence a concern for many types of interactive system. However, the central role of text in hypertext means that it is a significant concern for the hypertext designer. How significant probably depends on the user and the use to which the hypertext system or document is being put.

Users and uses

What of the users and their tasks? The users and others with an interest in the hypertext documents, the stakeholders, and the uses that they put such documents to are central to effective hypertext applications. These matters are dealt with in more detail in the next section; meanwhile, it is worth considering the users' relationship to the artefact. Who the user is should directly affect the form of the hypertext document; what the user is doing should also affect the shape of the hypertext document (Wright 1989). As Wright and others have indicated, what we refer to as reading is in fact a complex task, with a number of components. At each stage in the total reading process, different processes come into play: searching, comprehending, remembering - as well as the actual reading act (Wright 1990). There is also the question of what one is reading for. Wright identifies three different reading tasks purely within the area of hypertext for learning: reading for writing, reading for comparison, and reading for reference, and for each of these shows how the user interface could be tailored to suit the tasks involved.

2.5.3.3 The design process perspective: 'messy' problems

Problems with the hypertext product, the artefact, are only part of the overall design situation. Even though a product may be in itself successful, it may have emerged from a laborious and expensive development process. Having already discussed design problems in general terms, what problems can be considered as being specific to hypertext? It is arguable that the only design problems truly specific to hypertext are those pertaining to linking and the structuring of the hypertext network. Hypertext design utilises a number of different design skills and disciplines that exist in their own right, and pose their own hypertext-independent problems, in the production of hypertext documents.

Hypertext design problems are highly varied. They range from the question of how to structure the document in accordance with the knowledge held in the document, to ensuring a usable interface (Marchionini and Shneiderman 1988). They include the elicitation of knowledge about the user and their requirements, and about the tasks the user will perform with the document. They also include contextual aspects: for instance Glushko (1992a) identified a number of 'ways to make a hypertext project fail' such as missing design team skills, unrealistic expectations in terms of scale and readiness, and legal uncertainties regarding copyright and related issues. These issues are not centred on the hypertext document but on the total management of the design project. All these elements need to be managed by the designer.

An initial approach to structuring and understanding problems of this kind is to adapt a device from Duffy et al. (1992), originally applied to on-line help design, which shares with hypertext design a multi-disciplinary base of skills and concerns. They suggested that the design of on-line help involves applying the systems design process of analysis, synthesis and evaluation to two classes of what they called 'messy' problems: the problems of good document design and good interface design. Using this approach, hypertext design can be seen as involving the application of analysis, synthesis and evaluation to several further classes of 'messy' problems. It is reasonable to extend Duffy et al's structure and suggest that there is a class for each medium used. These classes will probably include graphic design, there may be animation, video and sound editing. How many classes are relevant in a specific case therefore depends on the nature of the hypertext and the media it employs. All of these classes are design subjects and disciplines in their own right, posing their own 'messy' problems. Then there are the problems of user interface design and of designing the hypertext linking structure itself.

A further set of problems fall less readily into this structure. These are associated with the type and purpose of hypertext, for instance information systems design and database design issues, or pedagogic problems. In the case of electronic prospectuses, the author suggests that these are considered as belonging to the domain of information systems, and the nature and implications of this are discussed at the end of the section. It must be remembered that methods appropriate to all of these need to be employed, and then co-ordinated and integrated, adding a further area of difficulty. This is likely to mean that some form of collaborative working is involved, which in turn poses further problems.

2.5.3.4 The 'messy' problems

The various problem areas have been studied in a large range of disciplines and subject areas. In some cases there is a very long history of study, and a substantial body of both theory and evidence. In other cases, the subject of study may be less established, but there is still much material. It is not therefore possible to review this work exhaustively in the present context, but before we conclude this discussion of the literature, some exploration of our present understanding of these activities is necessary for a further understanding of the hypertext design issue.

Central amongst these activities in any discussion of hypertext, whether generally or specifically with regard to electronic prospectuses is writing and document design. Graphics and illustration are also important and the associated skills of decoding and interpreting visual images and of designing them therefore become significant. As multimedia advances, it is increasingly likely that any electronic versions of the traditional prospectus will contain multimedia elements including video, sound and animation. The electronic dimension itself brings specific new challenges in terms of design problems and the nature of human-computer interaction.

2.5.3.4.1 *Writing, document design and information design*

Writing is of course central to hypertext. In the case of print media, centuries of familiarity have led to a large population of print users, both authors and readers, whose reading and writing skills are so practised as to appear intuitive (McKnight, Dillon and Richardson 1991). This appearance is deceptive. In the case of writing, the design of printed documents, from writing through to printed artefact, is a highly skilled process whose apparent ease is the product of those centuries of practice. Writing and publishing are the two parts of a total design process that has been perfected over several hundred years. This process has a number of common features, whilst still retaining a measure of flexibility that allows it to cope with a large range of document types.

Writing has been an object of study for many years, for instance in English literature, English language, linguistics and such applied disciplines as publishing, copy-writing and technical writing. All of these disciplines contain a wealth of useful knowledge about writing from a variety of perspectives. Although interesting not all of these are relevant to the problems of hypertext design. However, there is a body of knowledge exploring writing in the context of interactive systems, and this provides useful information for the purposes of this study.

Writing can in a sense be regarded as the opposite of reading, the coding of information into linear form as opposed to decoding it. In recent years closer attention has been paid to this process of encoding information, and to the elements of the writing process and what happens when people write (Sharples 1990). Sharples, drawing on much of this work, gives a hierarchical account of these processes. This includes at the highest level the two approaches to writing indicated by Bridwell-Bowles et al. (1987 in Sharples & Pemberton 1990). These are the 'Mozartian' writer, producing detailed plans before beginning to write, and the 'Beethovian', writing as a way of thinking through ideas and mixing the writing and planning activities. At a lower level, writers match specific strategies to tasks. Sharples identifies three common strategies: plan-draft-revise, draft-revise, and cut-and-paste. Within these strategies a range of more specific techniques may be applied to the retrieval, generation and organisation of material. Amongst these may be brainstorming, organising, materials gathering, outlining major sections, drafting text and revision. Sharples then extends this hierarchical model to suggest that writing is a complex cognitive task that involves a hierarchy of planning processes from document level to sentence level. There is typically not a step-by-step progression through the stages, but a cycle of planning, writing and revision. These tasks are interleaved. All the various strategies and techniques can be broken down into a set of basic operations. Text is typically reused, and texts are categorisable into types, such as 'article' or 'story'. Difficulties in writing arise because writing is an open-ended design task, with no fixed goal or formal intermediate states. It is also under-constrained: the possible ways of achieving a writer's goals are almost unlimited.

For the designer of hypertext, writing is made more difficult because of the new possibilities for structuring the raw material that arise from linking. Whereas previously one constraint on the author was the linearity of printed text, this is removed in hypertext. It can be hard enough to develop a good structure for a printed document; with hypertext, not only can otherwise implicit connections between items be made overt, but more than one structure can be imposed on a document. Hence there are pressures to adopt conventions to reduce such complexity such as specific structures, maps and conventions (Williams 1992).

Document design involves a process covering the writing and physical design of written materials. Like writing, three main stages of planning, drafting and revising can be distinguished in the process of document design (Wright 1981). The key concerns of document design, as distinct from related design areas such as writing and graphic design, are with the structure and format of documents. It is of course difficult to separate these areas - writing style and visual design are in many ways intrinsic to document design, and information design attempts to integrate these wider concerns. Document design and information design share with human-computer interface design a concern for the user, although in this context the term 'audience' is often preferred.

As far as information design for a hypertext electronic prospectus is concerned, the principal task is likely to be information seeking in some form. The literature on information seeking is widely scattered, although it is a central concern of information science. One attempt at an overview is that of Rouse and Rouse (1984). This proposes that information seeking is a process, involving changes in the nature of the information, and its value and relevance to the seeker, and that this is complicated by the ill-defined nature of the value of information. A further complication is provided by the blurring of the boundaries between information seeking and information processing. Cognitively, Marchionini and Shneiderman (1988) suggest that "users' information retrieval depends on the cognitive representation (mental model) of a system's features, which is largely determined by the conceptual model designers provide through the human-computer interface".

According to Marchionini (1992), information seeking is best seen as a form of problem-solving, composed of five main functions: defining the problem, selecting the information source, extracting information, articulating the problem, and examining the results. The information-seeking process begins with recognising and defining the problem. It is followed by source selection and problem articulation, where users formulate a query or determine their entry point to the information system. This is followed by examination of the results, and the extraction of information from the results data. The process is not in practice as linear as this description suggests, as the problem is subject to redefinition at all stages. However, Wright (1993) suggests that the psychology of problem solving is not directly relevant to information-seeking, and that little is known about the range of information-seeking strategies that people use, and about their selection of strategies.

Wright goes on to review a series of studies which suggest that electronic documents extend the range of reading strategies that may be required when working with electronic documents. These studies also suggest, according to Wright, that it cannot be assumed that users will have the necessary skills in information integration and in changing modalities from one representational form, such as words, to others, such as pictures and sounds. Wright further suggests that the implication of this for designers is that hypertext systems should be capable of supporting a wide range of alternative information-seeking skills.

2.5.3.4.2 *Visual design*

Visual design is an important element in hypertext design. It can be difficult to untangle from both document design and user interface design. Visual design concerns such elements as typography, the use of diagrams and illustrations, and the use of colour. Typography for print has been extensively explored, and there are a number of texts on this subject, for instance White (1988). Certain general principles and guidelines are commonly accepted. It is of course a design skill, an applied art, in its own right, although the rise of DTP is probably giving rise to a more typographically literate community. Typography for interactive systems is a thorny issue. In general people are still not keen on reading from screens, although the position seems to be improving as a result of WYSIWYG interfaces which more closely approximate the look of printed text (Gould 1988). In this context it is uncertain as to what extent the experience of print can be transferred to the screen.

The use of diagrams and illustration has also been examined, for instance by Duchastel (1980), although much of this work has been in an educational context. Again, there are specific skills involved.

Much work has been done on colour, usually from a psychological perspective. Guidelines have been suggested for the use of colour in interactive systems, for example Travis (1991).

Visual design presents several problems for the hypertext designer. First there are the specialist skills required. There are the limitations of the contemporary VDU screen, not only that used by the developer but that of the eventual user. Then there is the software. For example, as Kahn & Lenk (1993) show, control of white space is an essential ingredient in typographic design, yet most software does not allow full control over these elements, a particular example being the way most software does not allow the use of margins round text blocks, a device which enables the effective separation of text blocks and makes for ease in finding the beginning of each line.

2.5.3.4.3 *Designing for time-dependent media: sound, animation and video*

We now have a substantial history of listening to broadcast and recorded sound and viewing film, television and video. However, these media have only recently begun to appear in interactive systems. Their use in hypertext documents has a number of implications for the designer. All these media are dynamic, conveying information via change over time (McKerlie and Preece 1993). Such media are hard to work with because of their dynamic nature - they represent a moving target for the user of the interactive system. From a hypertext design point of view this manifests itself in difficulties with linking to specific sections of sound, video or animations, particularly whilst they are playing. In this context, the nature of the link anchor becomes critical (Trigg, in Streitz 1990). The problem is that the dynamic stream - sound, animation or video - cannot of itself support an anchor. Some form of interface object must be created to represent entities within the dynamic stream. Efforts to overcome this problem include the MIT AthenaMuse project's 'HotSpots' utility (Michon 1993). Even when a technique for embedding buttons within a video stream is developed, there remains the problem of representing the buttons to the user: in most cases, to see the video it is necessary for the button to be transparent and unlabelled. This raises the questions of how to alert the user to the presence of the button, and how to let them know what happens when they press the button.

Another area of concern is that of media integration, raised by Laurel et al. (1990). This refers to the phenomenon of 'media ghettos', where hierarchical structuring leads to a failure to access material in other media forms in a hypertext document. Such media segregation is heightened by the lack of conventions - as well as the technical and representational difficulties mentioned above - for linking to other media to parallel familiar print conventions such as footnotes and glossaries. It is further enhanced by the media biases which both developers and users bring to the authoring situation. For instance, Laurel et al. found that user trials of a multimedia database showed that users questioned the validity of video segments because they looked like TV and were therefore untrustworthy. Wright (1993) suggests there may be a cognitive cost to changes in media mode in hypertexts, citing work which found changes from text to auditory or visual information increased the necessity for users to reread text (Black et al. 1992, in Wright 1993).

Other general issues such as the specialist hardware and software requirements of multimedia are always considerations for the designer.

As far as the design process is concerned, multimedia elements present a new dimension for the designer in that the conventions that are used, for instance in computer science or in writing, for representing aspects of the design process, like data flow diagrams or outlines, do not exist for integrating multimedia elements with hypertext. A popular solution is the use of some form of structure diagram combined with story boards (Berk & Devlin 1991), but this is not extensively explored. Hodges and Sasnett (1993) have suggested that a theoretical basis for discussing and representing multimedia elements could be adapted from the film theory concepts of montage and mise-en-scène.

2.5.3.4.4 *User interface design as process*

We have already looked at user interface design from the point of view of the artefact. It is also necessary to examine it in design process terms, as another 'messy' problem for the designer to deal with. The key questions here are where and how user interface design fits into the hypertext development process.

The answers depend to a large extent on one's conception of the product design and development cycle. If one assumes that user-centred design is important - and most contemporary thought in HCI and user interface design suggests that it is - then it must be integrated into the hypertext design cycle. Although most writers on hypertext design are clear as to the need for attention to the user interface, exactly where user interface design fits into the design process is not extensively explored.

User-centred design aims to overcome the weaknesses of design processes that concentrate excessively on formal scientific and software engineering methods. There are several key techniques involved, although not all of these are used in all cases. They are all intended to involve the user in the design process: user analysis, task analysis, iterative prototyping. The latter implies that the systems account of design, analysis, synthesis, evaluation, that is widely employed in the design of interactive systems, has been rejected in terms of a model which more closely resembles what designers actually do, eg. Hix & Hartson (1993). That is, they spend a lot of time cycling back to earlier stages and revising their design in the light of tests to see how well the design fits the user's needs. In a sense, iteration is an extension of the traditional systems phase of requirements gathering, only it acknowledges that requirements cannot be gathered in advance of actual artefacts being shown to and used by the users. However, iteration also can be seen as a moving back of the traditional evaluation phase so that it takes place as early as possible. Summative evaluation is really too late to change a system; it can only improve the next one. Therefore evaluation must take place earlier and be formative in purpose.

Methods may also reflect the social context of design to a greater or lesser extent, looking at real users in real-world situations. A range of participative and co-operative design methods can be employed where it is felt that the user and other stakeholders should be actively involved in the design process. For instance, Wixon and Whiteside (1990) describe a process of 'contextual design' which involves users as partners in the development process, and Greenbaum and Kyng (1991) explore a number of participative design techniques and strategies.

Task analysis

In the broader context of software systems development, task analysis is clearly part of the analysis phase and its incorporation into the design process should therefore be unproblematic in terms of timing. In terms of a traditional systems life cycle model like the waterfall model, task analysis represents an element in the system investigation phase of analysis (Avison and Fitzgerald 1988). Problems with task analysis come in terms of selection of method, the appropriate level of analysis, and ways of representing the analysis.

Task analysis essentially means identifying the task(s) the user must perform at an appropriate level of detail. What is meant by task is not always as clear-cut as might be thought. For instance, Draper (1993) has suggested a number of task types. Hence task analysis methods vary widely, a number of techniques having been suggested and applied. These can be divided into three broad types, according to whether the analysis is concerned with the cognition, practice or logic of the task (Payne & Green 1989). Diaper (1989) explores a small selection of methods in some detail, and Jonassen et al. (1989) present a wider range of possibilities. The most commonly found type is hierarchical task analysis, in which the task is broken down into sub-tasks and operations. This is essentially practice-based. Cognitive task analysis (Johnson 1992) is oriented towards the application of cognitive theory to task situations by modelling the cognitive processes that are engaged in task situations.

Task in hypertext

Although the importance of task in hypertext design is recognised (eg. Wright 1989, Langford and Brown 1993), the business of exactly how to do task analysis for hypertext is somewhat neglected, with the exception of Jonassen et al. (1989) who identify several techniques such as pattern noting as especially suitable for hypertext task analysis, and Dillon (1994). Jonassen explores a number of methods for doing task analysis without looking particularly closely at the design context, whilst Dillon concentrates on task analysis in the context of a design framework for the creation of electronic text which involves the creation of a task model. It is not really Dillon's purpose to examine task analysis methods in detail, but the operation of a selection of methods is detailed. There is perhaps a problem in that the users of hypertext systems may be diverse, and their tasks may also be diverse. Tasks that are done using hypertext also tend to involve a substantive cognitive element, which is also less amenable to task analysis. Some suggested techniques for task analysis in hypertext design include pattern-noting (Jonassen et al. 1989) and structured interview with simulated verbal protocols (Dillon 1994).

User analysis

The designer of interactive systems clearly needs to know who the users are and what their needs and capabilities are. Designers are often reluctant to engage in this (Gould 1988), especially when faced with a heterogeneous population. This means studying users' various characteristics: 'cognitive, behavioural, anthropometric and attitudinal', and also the work which they wish to accomplish (Gould & Lewis 1985). This in turn involves answering a series of questions: what knowledge do they bring with them about using computer systems, about how computers work? What shared characteristics do they have demographically, in terms of professional role, and educationally?

A useful perspective for understanding the human aspect of user interface design is to think in terms of stakeholders, rather than solely in terms of users. Although users are very important, they are not the only people concerned with the development and final user of any system or document. The authorities who commission an artefact, those who pay for it, maintain it, are displaced by it or are otherwise affected by it are the stakeholders. Dealing with these various interests may involve a series of trade-offs between the various interested parties.

2.5.3.4.5 *Structuring information for hypertext*

One of the most perplexing questions for designers of hypertext systems is how to structure the information to be included. There are a number of aspects to this.

One consideration is, should the original document structure be retained, or should it be reorganised in a new purely hypertextual form? There is some debate about this. Some, for instance Glushko (1990) consider that retaining the original document structure is a good way of familiarising the user with this new medium, especially in cases where there is a long-established set of conventions for information presentation and access. Others, such as Brown (1989b), claim that firstly the very idea of hypertext contradicts the way in which traditional printed documents are structured, and secondly that to tie our hypermedia documents to these older forms is to deny ourselves the opportunity to use the new functions that hypertext offers. A framework is presented by McKerlie and Preece (1993), who suggest that there is a choice of authoring strategies. These are Carlson's 'shred and thread', where formerly linear documents are decomposed and the various relationships between ideas and concepts are elucidated, Nelson's organic evolution of hypertext trails through a body of documents which are linked by reader exploration, and beginning from scratch to develop a hypertext structure as the conceptual structure emerges.

It is likely that much depends on the form of the hypermedia document, in conjunction of course with the user population concerned and the task. Where there is a familiar and long-established form, there is often a very good case for retaining this format. A good example, explored by Glushko (1990) is that of the encyclopaedia, which can be seen as a hypertext document struggling to escape the confines of the print medium. The typical encyclopaedia structure and format is fairly standard. Users have an extensive understanding of this, based on its long use, and the information contained is already chunked and cross-referenced in a way that lends itself to electronic linking. Hypertext versions simply support and enhance this existing structure.

Where the document's structure is more distinctly linear in nature, such considerations do not arise. In this case, hypertext may not be a wise choice at all, as the opportunities for linking are minimal (Alschuler 1989). A classic case of this is of course the traditional novel, where the author's control as to what is revealed when is intrinsic to the form. More broadly, as a significant component of the creation of written documents is the linearisation of complex networks of material, it is arguable that existing documents may frequently prove to be unsuitable candidates for hypertext conversion, or at least only into a hypertext structure of a sequential type.

Essentially what matters is the underlying information structure. Information often has an existing overt structure that can be used as the base structure, for instance categories, chronology, location or continuum in terms of size, importance etc.

In broad terms, for hypermedia purposes, information can be divided into two types - sequential and non-sequential. When dealing with sequential information, the problem is how to break that information into chunks that work as hypermedia nodes.

If hypertext has a real advantage over more conventional means of holding information on computer, it is in the ability to link related subjects and concepts, to enable a progression from one chunk of information to the next that reflects semantic relationships. For hypertext to work most creatively it should enable the user to elucidate relationships between topics that have not been pre-ordained by the author. The difficulty comes when we try to translate the information structure into a hypertext structure. In theory the information structure should map directly into hypertext; in practice there are a number of factors that make this more difficult.

To make a hypertext structure demands linking. An important issue here is whether to link automatically or manually. Linking by hand is hard work. It usually means 'hard-wiring' the links. It means making decisions about each individual link to be included. It is suggested that hand-linked hypertexts benefit from retaining original document structures (Alschuler 1989). A large hypertext may be too big a task for hand linking. Automatic linking is a possible solution, but lacks the capacity to discriminate and may therefore link like to like on purely lexical grounds. The need to check may obviate any labour savings, although such checks could be combined with checks for dangling links. In any case such techniques are limited to those with access to extensive scripting and programming resources, not being available with commercial authoring tools.

Also a problem from the designer's point of view is how to develop a set of rhetorical conventions for linking. As mentioned earlier there are guidelines (eg. Landow 1989) which may assist the designer in deciding on conventions, but this is a problematic task, in particular because link representation may call for the use of typographic conventions which clash with those used to represent particular kinds of information, such as headings, sub-headings and other emphasised text.

2.5.3.4.6 *Information systems design*

Are hypertexts information systems? They appear to share certain common characteristics. However, definitions of information systems typically place them in organisational settings and roles (e.g. Avison & Fitzgerald 1988, Olle et al. 1993), and often give them certain dynamic characteristics: the output of progress reports or the processing of sales information. The typical information system is an information processing system. It is also clear that an information system need not be computerised, as Buckingham's definition indicates: "An information system is a human activity (social) system which may or may not involve the use of computer systems." (Buckingham, in Avison & Fitzgerald, 1988 p7-8). Hypertext can be used for a number of purposes, including information systems, and an electronic prospectus could certainly be described as an information system, although in terms of scale and central importance to the organisation it would be regarded as a rather minor one. This means that there is an extra layer of design issues to be dealt with and integrated into the design process, just as for instance a hypertext document for educational purposes would have to deal with pedagogic issues. Treating hypertext systems as information systems is actually relatively uncommon in the literature, both of hypertext and of information systems, although techniques from this and related domains are sometimes adapted for hypertext use (e.g. Perlman 1989).

At this point it is sufficient to note that information systems design is a complex and frustrating practice, the pitfalls and problems of which have been extensively discussed (e.g. Mumford and Henshall 1979, Checkland 1981, Avison & Fitzgerald 1988, Jayaratna 1994).

2.6.1 ELECTRONIC PROSPECTUS DESIGN PROBLEMS

Although there appear to be no problems relating to hypertext electronic prospectuses described in the literature, examination of hypertext design problems in general suggests certain areas of concern. One can expect that at least some of the problems associated with hypertext design will be apparent. For instance, there are a number of possibly contradictory interests that may need to be satisfied, from the HE institution authorities to prospective students. Other issues are hypertext-specific. Having an established paper counterpart gives the developer the difficult decision of whether to adapt the paper version or design from scratch (McKerlie and Preece, 1993). Prospectuses have to have regular updating and therefore must be designed to make this as easy as possible.

2.6.2 HYPERTEXT DESIGN

2.6.2.1 Existing design solutions

When the author first investigated this subject, only one model applicable to hypertext application design (as opposed to data models applicable to hypertext system development) existed, Garzotto et al.'s (1993) Hypertext Design Model (HDM) . This was developed into a fully fledged design methodology during the course of this research, Object Oriented Hypertext Design Method (OOHDM, Shwabe and Rossi 1995). There were only two published design methodologies specifically for hypertext, those of Van Vliet & Wilson (1993), and Perlman (1989). Although both presented interesting aspects, both lacked detailed techniques for managing the hypertext design process. Of the two, Perlman's was the more interesting, borrowing data flow diagram techniques from information systems design, but it had originally been intended as a methodology for doing research into hypertext, and as such did not appear ideally suited to the production of final designs. Van Vliet and Wilson also borrowed database design concepts, grafting these onto a description of the hypertext design process. Whilst this research was in progress, a further development in design methodologies for hypertext was Isakowitz et al.'s (1995) Relationship Management Methodology (RMM). This is characterised by a clearly delineated data model which is based on entity-relationship modelling and is a development of HDM and its successor, HDM2. The design methodology is described in some detail, and includes an overview diagram of procedures, which reveals the methodology to be mainly concerned with the middle phases of design, post-analysis and pre-evaluation. There are also some guidelines for entity-relationship design and navigation, and an example of the methodology's use for a higher education departmental handbook.

These methodologies tend to draw extensively from fields such as software engineering and database design. It is arguable that such techniques require at least some specialised knowledge, being intended in their original role for use by trained analysts. For the author whose primary purpose is to design a hypertext document this may be inappropriate.

There are also a number of descriptions of the total hypertext design process which have something in common with design methodologies, such as those described above by Shneiderman (1989) and Martin (1990), but these also tend to be short of detail. However, they do have the advantage of being comprehensible to non-specialists.

There are a number of general HCI guidelines which have some applicability to hypertext, for instance those by Brown (1988), and Smith and Mosier (1986), and guidelines specifically for hypertext development have been produced for instance Hardman & Sharratt's (1990) which were adapted from the previous authors' guidelines for hypertext, but these are largely artefact-oriented, and give very little consideration to design process and authoring aspects. Exceptions include Brooks (1993), but this is aimed at the technical authoring domain. There are interesting arguments against the use and value of guidelines and methodologies, which are considered in Chapter 4.

A wide range of other techniques exist to help software and user interface designers. These include software diagramming tools; support for documentation; project management tools; task analysis support, for instance Dillon's TIMS framework (1994); easy prototyping techniques; brainstorming techniques; graphic tools and visual programming software (Perez 1991), and techniques for exploring and displaying alternatives such as storyboarding. Most of these techniques have been applied in hypertext design - the literature has a number of references to them - but their use is rarely detailed, and many of the techniques are specific to particular software environments.

2.7 Conclusions

In general, the literature appears to show that, using the framework adopted in this review, the emphasis to date has been on the artefact and the user, to the neglect of the designer and the design process. The next area of concern should be, how does one make design easier yet still maintain quality of product? Current research concerns in hypertext show a move towards greater consideration of authoring and design aspects, and it therefore seems appropriate to examine this area more closely.

Many articles are evaluatory, or are concerned with functional or interface issues. Of course these cannot - and should not - be divorced from design, but one cannot help but feel that the author is still neglected, that there is still very little material that suggests ways to do hypertext effectively. There is a great deal of design advice in the literature, but much of this is scattered and contradictory.

To sum up, issues identified from the literature include the following key points:

1. Although the hypertext design process has been reported in a number of places, often on the basis of first-hand experience, very few of these represent systematic attempts to study the process. There is a large body of design guidance available, some of it codified into principles and guidelines, but much of this material is scattered between subject areas such as hypertext itself, HCI, technical writing and on-line documentation. The situation is further complicated by the extension of hypertext to include multimedia elements, and the consequent related design guidance.

2. Relatively limited attention has been directed towards support for hypertext designers. Design methodologies for hypertext are limited in scope and detail. Hypertext design guidelines to date appear to emphasise the quality of the resulting product rather than the design process. Many of them are derived from other areas, for instance HCI and are not specific to hypertext. There are still few texts which draw together useful design techniques in an easily applicable manner.

3. There is consequently a space for a hypertext design methodology that can provide a flexible yet helpful working environment for the less experienced hypertext designer (which may be most of us).

The implications of these points are developed in the next chapter, which explains the rationale and methods used in the study.

3: Rationale and Methodology for this Research

3.1 Aims: what this research seeks to do

The focus of this research was the hypertext design process, and its principal objective was to produce a hypertext design methodology. The following research aims were adopted:

3.1.1 OVERALL RESEARCH AIMS

The overall aim was to develop a hypertext design methodology. Electronic prospectuses appeared as a useful focus for the research. They cover a constrained body of knowledge, hence their size is similar to that of many actual hypertext documents, which despite the dreams of the hypertext evangelists are rarely more than a few hundred megabytes at most, and in the current phase are frequently distributed on CD-ROM. They are not fixed bodies of knowledge, but require revision, usually at yearly intervals. They are likely to incorporate more than one or two media types.

This could be sub-divided into more specific objectives:

1. To obtain a further understanding of the hypertext development process, particularly for hypertext electronic prospectuses, and more generally for other small to medium hypertexts for information provision.

2. To produce an accessible design methodology initially for the developers of small to medium hypertext electronic prospectuses, but capable of generalisation to the development of other small to medium hypertexts for information provision.

3.2 What areas of knowledge are relevant?

The literature review has already given some indication of this. The research territory is seen as the hypertext design subject area and a number of 'messy' problem areas, including writing and document design, visual design, designing for time-dependent media, user interface design as process, hypertext information design, and information systems design, were identified. Relevant disciplines and fields include human-computer interface design, information systems design, psychology, and computer science.

3.3 What we already know: implications of the literature review

It is clear that we already have an extensive knowledge of hypertext focusing on the user's problems and point of view. There have been a number of studies which examine hypertext use and the implications for best practice in the design of hypertext systems and documents of that use. Hypertext design as a process or a task is less well documented. The literature review showed that there have been relatively few systematic reports of the hypertext design process, although a number of papers and texts deal with various aspects of the design process. Support for hypertext designers tends to be oriented towards the attributes of the product rather than towards the actual process of design and authoring. There were no comprehensive or fully elaborated methodologies at the time of the literature review, although HDM appears to be developing in this direction with the development based upon it of the Object Oriented Hypertext Design Method (Schwabe and Rossi 1995).

There are a wide variety of hypertext applications and uses. The evidence of the literature suggests that it is unwise to over-generalise from specific task domains and user groups to hypertext generally. There is considerable force behind the arguments advanced by workers in the ergonomics and HCI fields for being highly specific in one's approach to the design of interactive systems in general. In the case of hypertext there is a huge range of potential document types, users and tasks. Hence there is substantial support for the idea that any design advice must be constrained to particular contexts of application. The question is, to what extent can one make generalisations about hypertext design? To put forward a design methodology or other kind of design guidance is effectively to make a generalisation or generalisations about hypertext design. There are a number of arguments put forward against such advice in the literature, for instance those advanced by Dillon (1994). However, the possibility exists that some limited but valid and useful generalisations may be made. Not all of the design process directly concerns the user, and not all design guidance need involve the user. For instance techniques for representing the hypertext design exist independently of the user's interest, even though it may be advantageous to use techniques which are comprehensible to the user. It is also arguable that design guidance relating to the final document can still be used with care despite reservations, and this question is considered in Chapter 4.

More generally, the literature review reveals that there are issues in the design of hypertext which require exploration. Hypertext authoring as a subject of study apart from hypertext systems and issues has been relatively neglected, although there are some accounts of the hypermedia design task. Yet hypermedia applications in general present a number of unique challenges. As is shown in the literature review, the design of hypertext applications is a complex task which requires the designer to become involved in a set of messy problems. Difficulties are compounded because those involved in hypertext projects may come from a number of different backgrounds possessing widely varying skills. If the people concerned have to work together, this range of skills and perspectives renders the process of collaboration more difficult.

What the literature rarely reveals is the authentic voice of the designer/author - articles like Brockmann et al.'s (1989) being somewhat rare. Although there are many reports on individual projects, some more detailed than others (eg. the SuperBook project, Egan et al. 1989) these seldom seem to reveal all the problems and the difficulties faced, preferring to emphasise specific and currently significant aspects, for instance architectural and other technical aspects (Meyorowitz 1986) or iteration with users (Egan et al. 1989). The more prosaic questions of how the designer coped with the design task and process are relatively neglected.

We do not know very much about electronic prospectuses. As a subject for hypertext they generate little excitement for their own sakes. However, they exemplify a kind of hypertext document which has been particularly neglected by the literature, which has tended to emphasise either large scale systems or small projects for research purposes. The typical text prospectus is of short book length. Hypertexts based on this material are therefore smaller than such items as CD-ROM encyclopaedias but still sufficiently substantial as to present a variety of design problems. As information systems they present aspects which may be generalisable to other hypertext information systems: their main user group consists of novice and occasional users; the task concerned is likely to be a fairly specific information-seeking task.

3.3.1 Why a hypertext design methodology is worthwhile

This information suggests that a complete detailed hypertext design methodology is a worthwhile goal, although it has been suggested that there is no need for hypertext-specific design methodologies (McKnight and Dillon 1995). It contributes another alternative to the hypertext methodologies so far developed. It forces a reappraisal of what is already known about hypertext design. It explores the extent to which the design of individual documents may be systematised. Even though the application of design methodology may prove to be premature, or even a cul-de-sac in the evolution of hypertext, a comprehensive description of the hypertext design task is worthwhile.

3.3.1.1 THE EXISTING DESIGN METHODOLOGIES

Existing design methodologies - as opposed to other accounts of the hypertext design process - are based on database design precepts. Many of these methods are not fully developed if compared to information systems design methodologies like SSADM for instance. Techniques are not described in full, and there are no documentation aids or other tools included. In fairness it should be said that the only widespread accounts of these methods occur in published papers, which quite probably do not represent the methods completely, but there is no indication that these methods are any more than outlines of procedures. The exception is HDM, which has been described extensively, and has a set of associated software tools under development (Schwabe 1992, Garzotto 1993, Schwabe and Rossi 1995). They all consist of general descriptions of a set of procedures, with a diagram showing the relationship of these procedures to each other. There are varying amounts of detail in the description of each procedure. The methodologies are considered below:

- **Asynchronous design methodology (Perlman 1989)**

This is quite detailed in overall terms, although apart from the use of data flow diagrams it does not outline any particular tools. Having originally been intended as a method of development for hypertext research, it assumes that development is an ongoing process and is based on an iterative model that makes it hard to know where to start and where to stop. There are of course arguments for this, which are dealt with in Chapter 7.

- **Relationship Management Methodology (RMM) Isakowitz et al. (1995)**

This is the most recent example, having appeared during the course of this work. This is the most detailed, and demonstrates how entity-relationship techniques might be used to specify the structure of information in a hypertext document. In this it resembles HDM, but it pays more attention to the later stages of development, such as text authoring and conversion and interface design.

- **Van Vliet and Wilson (1993)**

This is a rarely cited design methodology. It provides a fairly thin account of a set of procedures based on database experience and a few key sources such as Conklin (1987), and describes steps and tasks in hypertext design.

• **Hypertext Development Model (HDM) (Garzotto et al. 1993) and Object Oriented Hypertext Design Method (OOHDM) (Schwabe and Rossi 1995)**

As indicated in the literature review, HDM is strictly a model rather than a methodology. In accordance with this it does not attempt to provide procedures or techniques for all parts of the hypertext design process. What it does is support the description of the information structure of the subject matter of hypertext documentation. In this it resembles such common techniques from the database domain as the Entity-Relationship model.

HDM has more recently evolved into OOHDM, which describes a four-step process in which an object oriented model is built at each stage in the process. The steps are Domain Analysis, Navigational Design, Abstract Interface Design and Implementation. This methodology claims to differ from RMM in two key aspects. Firstly it emphasises navigational and abstract interface design issues, and secondly, by adopting the object oriented design paradigm, it claims to support the reuse of components.

Both HDM and OOHDM, and RMM are firmly rooted in systematic design concepts found in software engineering and database design. This renders them relatively difficult to use for designers from other backgrounds. It is questionable as to whether this approach is necessary or desirable for all authors of hypertext, and the contention in this research is essentially that an alternative approach, systematic yet accessible, is required.

3.4 Research design (1): what do we need to make a methodology?

Before considering the various alternative research approaches and methods, it is important to consider the general methodological orientation of the research. How is the research positioned with regard to philosophies and approaches to inquiry, and what kind of research is it: does it seek to describe, to explore or explain? The next section explores the theoretical context of the research.

3.4.1 THEORETICAL CONTEXT

3.4.1.1 Rationalism and pragmatism

To locate the research in terms of methodological orientation, it is necessary to consider the most common orientations. The traditional scientific orientation, Cartesian rationalism, embraces four approaches to computer system design: cognitive modelling; formal theory; methodology, and empirical studies (Coyne 1995). This approach has been extensively criticised in recent years, both with regard to computer systems design in general (Winograd & Flores 1986), and from domains such as information systems design (Hirschheim 1985). Such critiques can be broadly described as pragmatic. Their basis is in practice. They highlight the difficulties of knowing anything outside of context. Practice in the rational tradition is distinguished from theory and follows from it.

3.4.1.2 Positivism & Interpretivism

Another important categorisation is the common distinction between positivist and interpretative approaches. Galliers (1985) gives one version of this in which he divides the two approaches into scientific and interpretative approaches. The former covers such methods as laboratory and field experiments, surveys, case studies, theorem proof, forecasting and simulation. The latter includes subjective/argumentative approaches, action research, role-playing, futures research, descriptive-interpretative research and reviews. A development of this is Orlikovski and Baroud's (1991) division into three research paradigms: positivist (ie rationalist), interpretivist and critical.

3.4.1.3 Positivism and post-positivism and this work

If one applies the typical categories of scientific/positivist v interpretativism to the methods used in this thesis then one would say that these methods are quite conventional. But in some senses this work can be seen as having a hermeneutic/interpretativist perspective, as it concerns the interpretation of existing design guidance, the interpretation of people's design activities via case studies, and uses the author's own, highly subjective, experiences in electronic prospectus design as data. Yet methods such as secondary research and case studies are frequently regarded as part of the conventional positivist approach for instance Williams et al (1988) and Galliers (1992). The general approach taken in this study is not so much 'anti-positivist' but more what Hirschheim (1985) calls 'post-positivist' - doing something scientific but with an awareness of the rule of subjectivism, the need for context and interpretation. Such allowance for subjectivity is only right and proper in a domain which concerns the subjective experience of the designer, and the need to support the designer in that experience. In Thimbleby's (1990) phrase, '... science is necessary but not sufficient: rationalism is a powerful basis for understanding, but something more is sometimes needed.'.

3.4.1.4 The task-artefact cycle.

Carroll's (1989) notion of the task-artefact cycle represents an attempt to reconnect practice with theory. The task-artefact cycle is the idea that a designed artefact is, within the design context, the equivalent of a theory in science. When one designs any artefact (though Carroll is more concerned with interactive systems), one is effectively saying 'this is my theory about how this kind of artefact should be, and how it can be used". In other words, a designed object represents the existing state of the designer's knowledge. This knowledge is tested by using the artefact to perform the task it was designed to do, in other words, the task sets requirements for the artefact. In the process of use, the artefact may also redefine the task that it was designed to do - Carroll cites the example of the way that typewriting, word processors and DTP successively changed office tasks. So the task-artefact cycle is the cycle from tasks to requirements to artefacts, then to possibilities afforded by the new artefact, and back to the tasks performed. Carroll suggests that the task-artefact cycle could act as a framework for research and theory building in HCI, by employing design rationale, a detailed history of an artefact and its role, scenario-based design, a technique involving imagining an artefact in use before it is built, and representing this in various ways, for instance by storyboarding.

A common categorisation, which to some extent cuts across the research orientations, is to divide research into exploratory, descriptive or explanatory research, exploratory being the first familiarisation with the research topic and early theory development, descriptive being work on providing 'the context for adjusting and testing a theory', and explanatory being research that 'tests the extent to which a theory adequately represents the phenomenon being studied' (Williams, Rice & Rogers 1988 p46). This categorisation is not easy to apply to the context of this research, which contains some elements of all three categories but does not fall clearly into any one. A more useful alternative for this situation is the classification used by Iivari (Iivari, in Cornford and Smithson 1996), which divides research approaches into three kinds: constructive research, nomothetic research, and idiographic research. Nomothetic research is research using empirical data for hypothesis testing in order to produce generalisations from data, ultimately in the form of explanatory theory or laws. Idiographic research is research into cases or events, with the intention of understanding phenomena in context. This research can be seen ultimately as falling into the category of constructive research, research which is concerned with models and frameworks which do not describe any existing reality, but rather help to create one, and which do not necessarily have any "physical" realisation (eg IS development methodologies).

3.5 Research design (2) what are the likely sources of information?

This section examines where information on which to base a hypertext design methodology might be drawn from. Before doing this it is useful to consider what a design methodology is: what it might consist of and how it might function. We can do this by examining design methodologies for hypertext and for other fields, in particular information systems design.

3.5.1 INFORMATION ABOUT EXISTING DESIGN METHODOLOGIES: WHAT DO DESIGN METHODOLOGIES TYPICALLY CONSIST OF?

3.5.1.1 General examples

Design methodologies come in widely varying form. Differences occur in many dimensions: in terms of underlying philosophy and world view, scope ie. of which part of the design process is addressed, disciplinary basis, techniques and tools, and participants to name just some. A fairly typical example of a methodology for information systems design is SSADM (Nicholls 1987). This is a structured methodology, one of a family of methodologies which work on the basis of top-down decomposition. It is a hierarchical methodology consisting of a six-part set of procedures or 'Stages'.

The Stages are:

1. Investigation of Current System
2. Specification of Required System
3. Selection of Technical Options
4. Data Design
5. Process Design
6. Detailed Physical Design

(Nicholls 1987)

These Stages are normally preceded by a Feasibility Study. Each Stage is divided into Steps, which then are subdivided into Tasks. The relationship of Tasks to Steps to Stages is formally presented in the form of a Reference Manual which gives the navigational path through all these elements. Each Stage is a separate element with clearly defined start and end points, and as each one is finished it is subject to a Quality Assurance Review and must be validated and signed off before the next Stage can begin.

The design process is iterative, with outputs all open to amendment. Effectively SSADM is a highly formalised method of development, as the capitalisation above reveals. It is very detailed, and centres on the following of highly specific rules and guidelines. It makes use of specialised software tools. Its particular strengths are that by bureaucratising the development process it forces otherwise sloppy design procedures out and encourages and directs other participants to contribute to the design exercise. How appropriate is such an approach to hypertext design?

3.5.1.2 The hypertext examples

These have been described above. They are useful sources of information despite presenting a somewhat narrow database design approach to hypertext development. Even within this constraint they provide alternative versions of what hypertext design procedures may be like.

3.5.1.3 Some relevant themes

Design methodologies appear to have certain consistent features. In particular, they have an implicit or explicit model of the design task. They collect together a number of techniques for easing the process of design.

3.5.1.3.1 The systems approach

This has already been considered in the literature review. Design methodologies can all be seen as manifestations of systems thinking, and one in particular, Checkland's Soft Systems Methodology (1981) is a specific attempt to apply systems thinking to design methodology.

3.5.1.3.2 Information systems design methodologies (ISDMs)

In information systems, a design methodology consists of a set of procedures, techniques, tools and documentation aids (Avison & Fitzgerald 1988) intended to support the developers of such systems. SSADM, described above, is a commonly used example.

However, as was noted in the literature review, although the design of hypertext documents that present information has some characteristics shared with information systems, they do not have all the characteristics of information systems in the true sense. This being the case, what elements of ISDMs are relevant to hypertext information design? There is much variation between ISDMs, and it is difficult to generalise about them. However they all share a common concern with solving systems design problems, and the point really is, to what extent can hypertext design problems be seen as systems design problems? Hypertext documents do not share the same extensive data flows that are characteristic of information systems, but they do have a shared concern with the management of data, and the structuring of data. It seems reasonable therefore to assume that at least some of the concerns of information systems design are shared by hypertext design.

3.5.1.3.3 User interface design: prototypes and participation

User interface design utilises a number of methods and techniques which have been variously adopted for systems design. Some attempts have been made to make a completed design process based on user interface design needs, for instance, Hix et al's (1993) Star model for design. The lack of reliable theory and models for the design of hypertext documents suggests that user-centred design methods involving some form of prototyping are likely to be a necessary part of any design methodology for hypertext.

3.5.1.3.4 Software engineering

Software engineering utilises a number of conceptual approaches which possess attractions for the hypertext designer. In particular functional decomposition is a manifestation of the systems approach which has proved itself of great value (Sommerville 1992). This is typically a 'top-down' process, in which the whole system is decomposed into elements, which in turn are broken down into smaller elements until a level of detail is reached at which a small amount of coding can represent the task. The alternative is to start from the bottom and work up. This 'bottom-up' process works by specifying and coding the small details and assembling them together.

Again, how relevant is this to hypertext design? This partly depends on how one considers the relationship of a design activity like hypertext to software design. If one adopts an authoring perspective, concentrating on hypertext's relationship to printed texts, then the software engineering paradigm seems distant. If one concentrates on the information aspect, then software engineering appears insignificant. Hypertext design is not programming, although some hypertext documents in some systems require programming. Therefore hypertext design does not really have much in common with software engineering. But it could be argued, as for instance Sommerville (1992) does that software engineering is about software in a wider sense that includes analysis, maintenance and evaluation. If this is accepted then it is possible to see that software engineering is of relevance to hypertext design. It is certainly the case that software engineering falls within the ambit of systems design methods, and on those grounds alone it is arguable that there may be techniques and approaches that could be relevant.

3.5.2 WHAT INFORMATION DO YOU NEED TO MAKE A DESIGN METHODOLOGY?

A design methodology for hypertext could conceivably come from a number of data sources. These can be seen as dividing into two types: primary sources and secondary sources.

3.5.2.1 Primary sources

3.5.2.1.1 Personal experience

It could be based on first-hand personal experience, the accumulation of knowledge that comes with practice in designing hypertext. This appears to be the case with many of the ISDMs: their antecedents are in the authors' years of experience in systems design. This has the advantage of being direct and rooted in real design situations. On the other hand, it depends on the person concerned, their perspectives and biases, the nature of their design experience, their educational and employment background, and a number of other such variables. What works for one person may not for another.

3.5.2.1.2 *The experiences of others*

It could be based on the experience of existing hypertext designers. Questions here concern the extent to which generalisations can be made on the basis of this experience - to what extent can the lessons learnt designing one particular kind of system in one particular context be extended to others?

3.5.2.2 Secondary sources

3.5.2.2.1 *The literature*

The literature contains a wide range of material, from studies and reviews to collections of principles and guidelines, which may be analysed. This material contains much useful information, but has the disadvantage that as reported earlier, the literature does not contain much in the way of detailed accounts of the design process used in making hypertext documents. The focus is usually on some other aspects of hypertext. The literature does hold a lot of useful design advice pertaining to the hypertext artefact, but this is somewhat haphazardly scattered, and would require extensive work to collate and organise. Alternatively one may confine oneself to those reports which contain more overtly expressed and accessible design guidance. These include descriptions of the hypertext development process, some of them formalised into sets of procedures and methodologies, principles, and guidelines.

3.5.2.2.2 *Other sources*

Another potential source is represented by the hypertext documents themselves. These have beneath the surface layer the evidence of construction, in the form perhaps of programmers' notes, in the style of programming, perhaps in the way in which screen objects are constructed and layered.

3.5.3 SO WHAT MIGHT A HYPERTEXT DESIGN METHODOLOGY CONSIST OF?

For this thesis to be genuinely useful, it should contribute more than the existing methodologies and other design guidance. The initial conception, based largely on the author's prior experience, was of a set of procedures which would incorporate a task and user analysis procedure, a hypertext structuring procedure, and an information design procedure. Other than this, the conception was somewhat vague. The literature leads one to contemplate this in more detail. On the basis of information systems methodologies, one would expect an overall set of procedures, perhaps but not necessarily in steps or stages. Coupled with this would be a set of techniques and possibly software tools. These would be more closely tailored to the design of hypertext documents. Some techniques for integrating task analysis into hypertext design and translating the results into hypertext structures would also have a place. The existing hypertext methodologies tend to concentrate on general procedures and could usefully be supplemented by some relevant and tailorable interface design guidelines for hypertext, including some guidance on how to mange the use of such material effectively. Documentation aids for some of the tasks identified in the literature as awkward such as link management, could be included.

This is somewhat vague, but in many ways this vagueness reflects the broader experience of design. The development of technology is an iterative design process, rather than a process of transfer from scientific principle to technological application (Carroll, Kellogg & Rosson, 1991). This applies as much to the techniques which form a part of the technology in question as to the apparatus or artefacts which result. New inventions are built on the backs of older and lesser designs, and new ways of doing things, new techniques, emerge from a process of iteration. Knowing this imposes a tension upon research of this kind: much of what is done may seem contingent and ad hoc. The trick is to balance a need for acceptance of vagueness and a willingness to try things out with a need for a structured and logical approach to the development of techniques which has some scientific credibility.

3.5.3.1 Models of design

Most methodologies have an underlying model, although this might not be overtly acknowledged. For instance, RMM (Isakowitz et al. 1995) is clearly based on the RMDM model, which is viewed as integral to the methodology. This suggests that it may be useful to produce a model of hypertext design. This seems like the best thing to base the overall set of procedures on. How would this work? In Isakovitz et al.'s case, the model is specific to the methodology, and is not an attempt at a description of what happens in hypertext development. An alternative strategy is to make a model of hypertext development which attempts to describe the typical development process and then to use that as the base for the procedure. But it is necessary to be careful to distinguish between descriptive and prescriptive models here. A description of what happens in hypertext design may not provide an appropriate model for what to do in hypertext design.

3.5.3.2 Overall procedures

The overall procedure, the organisation of design phases, has a number of manifestations, from stepped models of process such as the 'waterfall' model to asynchronous models such as that underlying Perlman's design methodology (1989).

3.5.3.3 Techniques , guidelines and tools

3.5.3.3.1 Techniques

A large part of a design methodology is the techniques it adopts. These include task analysis techniques, data analysis techniques, methods of notation and representation and diagramming techniques.

3.5.3.3.2 Guidelines

There are already extensive sets of guidelines for user interface design, and these have been adapted for hypertext design, for instance Hardman and Sharratt (1990) adapted those of Smith and Mosier (1986) and C.M. Brown (1988). Such guidelines are not without controversy. They have been criticised for several reasons, many of which were well summed up by Dillon (1990). They tend to suffer from over-rigid interpretation at the expense of user-specific, and task-specific requirements. Sometimes they become set in tablets of stone over time, potentially risky in a fast-changing field. Sometimes they are simply wrong. However it is the author's contention that administered cautiously these aids to design may be of value. These issues are paid further attention in Chapter 4.

3.5.3.3.3 *Tools*

Tools, as defined by Avison and Fitzgerald (1988), refers to software-based support for design. This is an interesting area, and a number of hypertext design tools have been suggested and produced (Perez 1991). However, this subject was considered as being outside the scope of the present study. Some further comments on this topic are made in the concluding chapter.

3.6 Research design (3) What are the best methods for getting the required information?

3.6.1 CHOOSING RESEARCH METHODS: A REVIEW OF ALTERNATIVES

Given the methodological orientation described above, and the range of possible data sources, no one research method appeared totally adequate. Consequently, a pluralistic approach was considered appropriate for this research problem, with a number of options being considered and eventually three methods adopted.

3.6.1.1 The options

Having decided that the research in question is constructive research, what research methods are appropriate to this type of research? Williams et al (1988) describe a number of possible research methods appropriate to the study of new media:

3.6.1.1.1 *Positivist approaches*

These include such methods as experiments, quasi-experiments, surveys, longitudinal, field studies, archival research, forecasting, content analysis and case studies. Such methods have been extensively criticised in recent years. In particular they are singled out for ignoring the way in which social contexts are constructs not given realities. They do not take into account the uniqueness of human situations and the values inherent in researchers and the methods and theories they adopt.

3.6.1.1.2 *Interpretative approaches*

Sometimes known as anti-positivism or naturalism, and increasingly common in information systems research. Methods associated with this approach include ethnographic methods and action research.

3.6.1.1.3 *Methods used in relevant areas*

Common methods found in the hypertext, information systems, HCI and other related literature include hands-on practice, the design and making of artefacts, laboratory and field experiments, mathematical modelling, and case studies.

3.6.1.2 Choice of methods

Within this framework, a number of alternatives could be considered. The key point is that the research was pragmatic in outlook: the overriding rationale being to find information that helped the construction of the methodology. It was felt necessary to have some data about what existing designers of hypertext systems do, and it was felt that this data should come from experience with similar systems. However, the relative novelty of the area suggested that there were unlikely to be sufficient instances to use survey methods. Participative and action research approaches were not felt to be practical. Hence the idea of a case study approach was adopted.

The project as originally envisaged intended the creation of a prototype electronic prospectus followed by evaluation and the development of a generalised model or methodology on the basis of this work. As explained above this method is common in the relevant disciplines. However, this approach presents certain problems. Although it is possible to make analytical as opposed to statistical generalisations on the basis of a single case (Yin 1994), one is faced with the need to deal with one's own biases where that case concerns a system of one's own design. Consequently it was decided to explore other methods of supporting data gained in this way. Quantitative methods were seen as likely to be inappropriate given the relatively small number of hypertext developers and the difficulties involved in finding a representative population. Experimental methods were ruled out on similar grounds: it would have been very difficult to find sufficient skilled participants with the time to become involved. Participant observation was also regarded as impractical. Case studies were suggested as a way of giving an alternative set of perspectives. It was also considered that the considerable body of knowledge that already existed could not be ignored. Some form of secondary research was therefore necessary. The problem was how to include this material without a disproportionate amount of labour. Much of this material has already been summarised and codified to some extent in the form of design guidance such as guidelines, principles, models and methodology. This material could be collated and organised, and the resulting condensation be incorporated into the proposed methodology.

The result was a combined methodology which has its precedents both in the social sciences generally in the notion of triangulation, which uses the idea of cross-checking inferences from more than one data source (Denzin 1994), and more specifically in the fields of HCI and information systems, for instance triangulation was used by Trauth & O'Connor, using a set of qualitative methods (1991). A similar approach is advocated by Galliers (1991). Such an approach has the advantages of dealing more effectively with complex situations, and the changing requirements that emerge during a research intervention (Mingers 1996).

The three research processes which were adopted were the construction of a prototype electronic prospectus, the collation and classification of a range of existing hypertext design guidance, and case studies of design teams working on electronic prospectuses. This would be followed by the construction of a hypertext design methodology.

3.6.2 THE METHODS CHOSEN

The three-way approach adopted therefore consisted of the following elements:

3.6.2.1 Review and classification of existing design guidance

This comes under the heading of archival and secondary research in Williams' review of methods. Such research involves the use of materials including published methodologies, principles, guidelines, rules and other design guidance. Analysis would consist of a classification or taxonomy of the material, to identify its relevance to various aspects of the proposed methodology.

3.6.2.2 Prototype

The design of a prototype document, involving an exploration of the design situation including task and user analysis as appropriate, and the design and construction of a prototype or prototypes.

3.6.2.3 Case study

It was hoped to find an example or examples of electronic prospectus design, and to study how the project or projects were conducted using a case study methodology designed to allow generalisation from single cases or linked cases as appropriate. The case study method was seen as appropriate because if conducted correctly it enables generalisations to be make from a very limited number of data points.

3.6.2.4 Integrating the methods

Having decided that this is an appropriate approach, it is then necessary to consider how the three methods should be integrated together, firstly in terms of the sequence that they should be performed in and secondly in terms of the integration of the data into a methodology.

3.7 Research design (4): how does one construct a design methodology from this information?

3.7.1 RELATIONSHIP OF THE DATA SOURCES

The initial work concentrated on the review and classification of the various design guidance elements found. This began with the construction of a table summarising and classifying a number of design guidance sources. This was followed by the construction of an early prototype of an electronic prospectus , partly informed by this material. These provided some loosely formulated data on which to base the case study protocol, in particular providing ideas for appropriate interview questions. The whole process was to a large extent an iterative process, in that the case study then provoked a re-evaluation of the other parts, and future work on these. The original prototype was discarded and a further prototype constructed. In the meantime, the guidelines were imported into a database and reclassified. During the period of the project, the question of what a design methodology for hypertext might be like was also addressed, and a simple design methodology for hypertext development was produced by synthesising a number of existing hypertext methodologies and similar design guidance to produce a model of the hypertext design process. Whilst being somewhat over-dependent on conventional structured design methods, this proved useful in providing a framework for the construction of case study interview questions, and to a limited extent informed the construction of the second prototype.

The prototype later functioned as a 'test-bed' for the various methodology elements, with a range of techniques and procedures being tried out using the prospectus as raw material.

3.7.2 PUTTING TOGETHER THE DESIGN METHODOLOGY

Having made some tentative decisions about what might be included in a methodology, how does one go about constructing a design methodology? The literature is surprisingly reticent on this matter. There are many accounts of methodologies, but the process of genesis is rarely described in any detail. For instance Cutts' (1991) generic structured system analysis and design methodology is credited to a basis of industrial, research and consultancy experience, and is closely related to SSADM, having involved work with Learmounth and Burchett Management Systems, who were closely involved in developing SSADM for the Central Computer and Telecommunications Agency. It is clear that SSADM has a long history, going back to the NCC's early efforts at systematising the information system design process, but the details of this development process are not given in Cutts' book. This is fairly typical. A more transparent account is found from Avison and Wood-Harper (1990), the developers of MultiView.

Because of this lack of detailed information, it is most productive to simply look at what the various methodologies provide in terms of design support. This is more transparent, being the subject of explanatory and instructional texts like that of Cutts (1991), although the fact that information systems design methodologies in particular are often commercial products costing thousands of pounds may make it difficult to see the artefacts associated with them. This design support varies according to the paradigm that is adopted. For instance, Soft Systems Methodology contrasts with the highly bureaucratised approach of SSADM in assuming that there may be a number of different representations of the truth in describing a system, whilst SSADM and other structured methodologies assume that there is only one true representation of the system situation (Jayaratna 1994).

Given this lack of a firm and specific basis for methodology development, it is necessary to examine the various possibilities. A number of sources were considered. Exactly how a hypertext design methodology would emerge from these various sources of data was not envisaged in detail at first. The initial assumption was that once sufficient data was obtained, some generalisations about appropriate procedure could be made, and an initial procedural outline developed. This could then be fleshed out with more detail, and appropriate techniques described and added.

The first efforts at making a methodology took place relatively early, in the form of a set of exploratory diagrammatic representations of sets of procedures, a 'top-down' approach to the problem. These tended to pay insufficient attention to the specific circumstances of hypertext design. It was clear at this point that a more systematic approach was required. The initial 'top-down' approach was combined with a 'bottom-up' approach. A model outlining the sequence and relationships of the procedural stages was compiled from the various data. At the same time, various techniques for localised design tasks were developed or borrowed. These were identified as necessary in the first instance from experiences with the electronic prospectus and from difficulties identified in the literature. These techniques could then be integrated with the procedural model to give a methodology. The process of developing the methodology is described and discussed in Chapter 7.

3.8 Problems and limitations: the moving target: issues of time and change

The pace of change in interactive systems is rapid. This perhaps explains the tendency for earlier work in such domains such as HCI and information systems to be theory-orientated when this was not particularly constructive. When phenomena are transient, they are less attractive objects of study (Davis 1992). This effect is observable in this study in that it is necessary to avoid the vagaries of particular systems when making generalisations. This is obviously problematic and requires a degree of suitable reflection on the part of the researcher.

3.8.1 THE ROLE OF DATA

This study does not involve the gathering of a large amount of data. This is justifiable on the grounds that a considerable part of the research is constructive and synthetic, involving the synthesis of existing data and new data. The main point is to gather enough data of both kinds to provide sufficient evidence for the construction of the design methodology. Whether this has been achieved is more substantively addressed in Chapter 8.

4. Design Guidance: a Taxonomy

This chapter reports efforts to utilise the experiences and observations of others in a design methodology by producing a compilation and taxonomy of design guidance. It examines the method used: the source materials, and the rationale for their selection, the characteristics which provided the basis for classification, and the analysis to which they were subjected. Finally, the results of this section of the research are summarised.

4.1 Rationale: the role of the taxonomy in the overall study.

The experiences and observations of others are manifested in a wide variety of forms ranging from reports of design experience through the outcomes of a large number of investigative efforts, including experiments, various forms of observation, surveys and other methods. Much of this material was reviewed earlier in conventional fashion as part of the literature review. However, a considerable amount of the published material in relevant disciplines includes more or less detailed recommendations for proceeding with design. It was considered that this material had to be taken account of in more detail in order firstly to properly evaluate its potential role in a prospective design methodology, and secondly to enable its most effective use in a methodology.

It was therefore considered necessary to provide at the very least a clear and succinct presentation of all the relevant guidance. At its most rudimentary, this would consist of a compilation and listing of all this material. However, this gross unrefined listing would present certain problems. In the first instance it would contain considerable repetition, which would not always be immediately apparent due to differences of expression and terminology. Secondly, an unstructured listing would be cumbersome and difficult to use. So some degree of inspection, refinement and classification was clearly desirable. The procedure via which this was accomplished is described in more detail below. Meanwhile, it is necessary to examine firstly the role and form of, and secondly the value of the various kinds of design advice in the relevant design domains.

4.1.1 THE NATURE, ROLE & VALUE OF DESIGN GUIDANCE.

Principles, guidelines, rules, standards, methodologies, models and other forms of design advice are common throughout the area of interactive systems. Their value is often contested, yet their persistence suggests that they have some value. What that value may be is discussed below. First, some investigation of the kinds of design guidance and advice is required. There is a wide range. Theories, models, frameworks, principles, guidelines, rules, rules-of-thumb, heuristics, standards, methods and design methodologies are all terms used to describe various kinds of design guidance. It is often not clear what is meant by each of these terms: they may be used interchangeably by the various authorities, precise definitions are not often forthcoming, and there are often overlaps in the areas covered by terms.

A starting point for investigation of the various elements involved is to examine more closely Shneiderman's (1992) taxonomy of design guidance. This describes four kinds of design guidance: 1) high-level theories or models; 2) middle-level principles; 3) specific and practical guidelines and 4) strategies for testing. This taxonomy is initially attractive. It separates out key areas of design advice effectively, and appears to provide a good basis from which to consider the design advice types found, although it fails to consider design methodologies or frameworks, which should presumably be considered with the high-level theories, or rules and standards. To some extent these taxonomic questions are a red herring. The important distinctions are to what extent the advice concerned is theoretical or empirical in basis, and whether the advice is general or specific in its applicability.

A closer examination of Shneiderman's classification illustrates this. In this taxonomy, high-level theories and models provide an application-independent context for the understanding of design issues. Shneiderman identifies three sub-divisions in this category: theory, taxonomy and models, to which we may add methodologies and frameworks. Theory is divided into the familiar categories of explanatory and predictive theory. This can be queried - can there be any kind of predictive theory which lacks the power to explain? - but this is not problematic for our current purpose. In the realm of interactive systems, explanatory theory is useful for the observation and description of behaviour and activities, design conception and comparison, whilst predictive theory may enable the comparison and evaluation of proposed designs. Taxonomy is considered by Shneiderman as a kind of theory, a way of ordering complex phenomena, in order to enable comparison, and guide design. Some confusion creeps in because the term 'theory' is used by Shneiderman in two senses: as a catch-all for all of these types of conceptualisation, and as a specific term - so 'theory' awkwardly becomes a subset of theory. Several examples of models are given, making it apparent that Shneiderman's definition of the term model is quite loose (see below for discussion of this and related matters).

The second level, middle-level principles, are seen as a way of facilitating understanding of the user and the situation of use in order to make design choices. Although it seems intuitively right to separate out a lower level of principles, it is hard to see from Shneiderman's description exactly what defines these. The 'underlying principles of design' that are presented here (Shneiderman p72-4) are described as being 'derived heuristically from experience', and require to be 'interpreted, refined and extended for each environment'. Guidelines documents are seen as ways of ensuring consistency across groups of designers, recording practical and empirical knowledge and providing 'useful rules of thumb' (Shneiderman p78). Again there is some doubt as to what is meant by a guideline: the 'high-level objectives' quoted from Smith & Mosier (1986) seem remarkably similar in kind to the earlier 'underlying principles of design', albeit applied to a narrower domain. Shneiderman (1992) does not consider such forms of advice as rules and standards. Shneiderman's final category is that of prototyping and acceptance testing. This concerns the setting of criteria for acceptance of a system, and the ways of testing for achievement of such criteria.

Despite the uncertainties mentioned, Shneiderman's taxonomy provides an initial framework for analysis. Smith (1988) examines the levels of guidelines, rules and standards in more depth. Again, the perspective is that of user interface design. He considers four alternative forms of design guidance: design standards, guidelines, rules and algorithms. Design standards are generally stated requirements for user interface design which are imposed formally by decree, contract or some other binding way. Guidelines are generally stated recommendations, which are open to selection, modification and adaptation by designers. In other words, they have no mandatory force. Design rules are seen as design specifications for a particular system application, which are specifically stated so as to require no interpretation or adaptation for that system, for instance rules about the positioning of titles or windows at specific places on the screen. Design algorithms are computer routines or programmes that embody specific design rules, as applied in user interface management tools.

Key points here are variations in the extent to which these types of design advice carry official force, the extent to which they are publicly available, and the degree to which they apply specifically or generally. A further important point is the extent to which tailoring is required: the degree to which selection, interpretation and modification is required. The point at issue here relates to the more general one considered below of the generalisability of design guidance. Smith sees general standards and guidance as being in need of translation into more specific design rules and algorithms before they can be applied. More generally, one of the problems of design advice is that it may often require tailoring to suit circumstances. This is reflected in the way in which different kinds of guidance are expressed, with standards being described in such a way as to admit no possibility of variation, and frequently with the accompaniment of procedural rules for exemption from, or alteration of particular standards. Guidelines on the other hand often provide examples of the use of the guidelines in specific circumstances. This is the format adopted in Smith and Mosier (1986), for instance, or the HyperCard stack design guidelines (1989).

This gives us some greater understanding of the structure of the realm of design guidance. Some problems in the elucidation of structure still remain, and these are considered further in the section on the taxonomy methodology later in this chapter. There also remains the key question of the value of design guidance. How useful or worthwhile is design guidance in its various manifestations? The next section addresses these issues, beginning with high-level guidance.

4.1.1.1 The advantages and disadvantages of design guidance: theory, models, frameworks, methods and methodologies

Theory manifests itself in design in a variety of ways. Shneiderman (1992) notes several examples of the use of theory as a guide to the design of interactive systems, for instance Foley & van Dam's Conceptual, Semantic, Syntactic and Lexical model of the interactive system, and his own Syntactic-Semantic model of user knowledge. Other examples include Dillon's (1994) framework for aspects of the user of electronic texts, which attempts to represent the various human behaviours and cognitions surrounding the reading process. There is a long history of attempts to systematise and impose method upon design in general which was discussed in Chapter 2.

Some general points about the role of theory in the design of interactive systems should be reconsidered before examining particular aspects such as models, methods, methodologies and frameworks. There is no doubt that such models and theories are attractive: the existence of sufficiently well validated models of, for instance, users would enable the construction of systems to be streamlined. The evidence to date is that designers and companies are reluctant to utilise techniques from the HCI/human factors area involving successive iterations on the grounds of inconvenience and expense (Hannigan & Herring, in Dillon 1994). A further attraction of theory is that the generalisations of theory may help to simplify otherwise intractably complex problems. But despite the attractiveness of theoretical approaches there are considerable difficulties, many of which are raised by John Carroll's collection of articles (Carroll et al. 1991). Carroll's basic case is that the efforts of HCI practitioners to find an applied psychology that could adequately support the practitioners of computer system design has failed because 'basic science provides uncertain and indirect support to practical endeavours.' (Carroll 1990, 1991). Carroll's case extends more widely than a critique of psychological theory as applied to HCI: he is also concerned with critiquing the methods of experimental psychology as applied to HCI. However, his critique of the role of theory, and his alternative approach, is relevant here. Essentially this is that applying psychological theory obtained in specialised circumstances is inadequate for the HCI context. A key point is that much theory rests on simplification, whilst recent work in HCI suggests that the products of design can only be properly understood in the context in which they are used (Suchman 1987, Wixon et al. 1990, Winograd and Flores 1986). However, as Dillon points out, it is impossible, according to contemporary accounts of the scientific process (e.g. Chalmers, in Dillon 1994), to find empirical work that is completely untainted by theory. Dillon suggests that the reality is that practitioners (he is referring here to human factors practitioners, but there appears no reason why this point does not generalise to interactive design practitioners in general) find themselves somewhere on a continuum between frequent empirical iterations and theory-based design. The implication is that frameworks and models have a place in HCI as a way of deriving appropriate initial designs. This reduces the number of iterations required to an acceptable level, and evaluations of prototype systems based on such frameworks or models effectively also evaluate the frameworks or models themselves. However, Coyne (1995) raises a number of further objections to the utilisation of theory, method and models in design.

4.1.1.1.1 *Methods and methodologies*

General design methods: advantages and disadvantages

With design methods and methodologies similar criticisms can be applied. (The author follows Jayaratna (1994) in adopting the term 'methodology' because of its widespread use in information systems, although as he points out the term 'method' is more appropriate.) Design methods and methodology are obviously features of a large number of applied disciplines. Attempts have been made to produce generic design methodologies, and these have been briefly examined in Chapter 2. The problems associated with these are relevant here. As Coyne (1995) points out in the general context there are no reliability criteria. Following a method will not produce the same outcome regardless of situation and personnel, as is the case with an experimental method. There are also, apparently, no formulaic theories of design. There are a number of generalisations, but these do not fit well together.

Information Systems Design Methodologies (ISDMs)

More specifically relevant design methodologies to the context of interactive systems are information systems design methodologies, and some similar criticisms are encountered here.

An extensive range of information systems development methodologies exist. It is hard to consider these within the current context. The classification and comparison of ISDMs is complex and chaotic, and is therefore a subject of study in its own right. However, it was felt that ISDMs should be considered with regard to the taxonomy, even though comprehensive inclusion was not possible, for two principal reasons. Firstly as the overall concern of the total study is with hypertext as a provider of information, it is logical and appropriate to make some examination of a field which is so closely concerned with information. Secondly, at the level of features, many ISDMs undoubtedly contain features which may be adaptable for the design of certain kinds of hypertext applications. The hypertext design methodologies described above are themselves clearly influenced by ISDMs and database design.

Like the other forms of design guidance, ISDMs also pose problems of credibility. Avison & Fitzgerald (1988) indicate two main areas of difficulty as far as ISDMs are concerned. First, they quote Checkland's assertion (1987) that it is not possible to prove that the success or failure of any information system can be attributed to the methodology employed. This has important implications for evaluation. Each information system is a product of the unique combinations of developers and environments in each design situation, which makes repeatability or the use of controls impossible. Secondly they point out that methodologies are moving targets: the technology is continually changing, and the discipline of information systems is so new as to be lacking consensus on many issues. This leads to a number of more specific problems in comparison. There are often documentation and versioning problems. Which particular version of a methodology is under scrutiny? Many ISDMs are commercial products, which are frequently expensive and not available for open research. The available documentation does not always represent how the methodology is applied in practice. The administration of particular methodologies by developers is not consistent. Another key issue is terminology. There is very little consistency in the application of terms in information systems development. This makes it very difficult to be sure what a methodology actually consists of without actually using the methodology. Of particular interest here is the lack of consensus both on issues and terminology. This makes the business of selecting and applying categories for classification more difficult in that there are no agreed categories or definition of terms that might be used for categories.

4.1.1.1.2 *Models and frameworks*

Models and theories can be hard to distinguish. In most cases models are ways of manifesting theories in ways that make them comprehensible. In the hypertext domain, the term model is used in two ways, both of which involve theoretical generalisation, and which therefore are subject to the same kinds of criticisms already discussed. The first use is to describe abstractions of hypertext application structures. Interest in this kind of model stems from a need to find ways of defining hypertext structures that are not application- or system- dependent. At the most basic level, this is a matter of finding a common terminology and semantics that enables comparison of systems (Grønbæk & Trigg, 1994). Beyond this is the awareness that, as interest grows in adding hypertext functionality to a range of applications, and in providing 'virtual documents' that are capable of use by such applications, it becomes necessary to be able to define interchange formats by means of a formal model of hypertext structure (Frisse & Cousins, 1992). Examples include the Dexter reference model, (Halasz 1990) the Hypertext Design model (HDM) (Garzotto et al. 1993), and Campbell & Goodman's 3-level model (1988, in Nielsen 1995) The second use of the term is to describe hypertext development processes in the abstract. An example of this is Schwabe & Rossi's Object-Oriented Hypermedia Design Method, which attempts to describe generically the process of building a hypertext application, and appears to fall somewhere between model and methodology (Schwabe & Rossi 1995).

Models seem to be taken at face value in the hypertext world at present. Critiques are addressed at the level of the structure and terminology used in models, rather than concerned with the value of models for hypertext. This is probably because of the predominance of models of hypertext structure rather than process.

A related approach is the framework concept. The term 'framework' is used by Dillon in the manner suggested by Whitefield (1989, in Dillon 1994) to refer to a generic representation of important aspects of the user, as opposed to a model, which is seen as a specific representation of those aspects in the context of a particular task. (A model may of course be more broadly defined however.)

4.1.1.1.3 *Guidelines, principles and rules*

Smith (1988), as befits a man who was responsible for one of the most widely cited sets of interface design guidelines, is largely positive about design advice in general, but others are less convinced. Thimbleby (1990) draws attention to some of the paradoxes, contradictions and other problems that are apparent in the use of principles and guidelines for interface design (he uses the terms interchangeably). He notes that many principles have been derived from small studies that may not generalise well to other systems developed at other times, in other places and on other equipment. Such principles should not be over-generalised by inexperienced designers, yet it is the whole point of principles to be used in this way by precisely those inexperienced designers. Furthermore, principles tend to be either trivially general, or difficult to implement. The vaguer the principle, or the more trivial, the easier it is for the designer to implement. Substantive principles may mean that any changes necessary to follow such principles result in in-depth changes that designers are loath to implement (Thimbleby 1990). In general, guidelines tend to be application-dependent and user-dependent. They apply in some cases and not others - there are frequently conflicts between guidelines and contextual conditions. In many cases there are trade-offs to be made between functionality and usability guidelines.

In general Thimbleby sees guidelines and principles as being inadequate in themselves, although they help to promote consistency between systems developed by different companies or designers. His solution is to promote guidelines which are accessible to both designer and user, which are integral to the user's developing mental model of the system.

Further criticisms of guidelines and principles have been made by Dillon who considers that ' no theoretical models or formal guidelines exist that can even approximate the quality of information obtained from observing real users interacting with a system' (Dillon 1989). This may well be true, but as he himself points out, some people like guidelines, and as Nielsen, a great advocate of heuristics, says, testing with users can be time-consuming and expensive (Nielsen 1989). And Dillon is sufficiently flexible to consider that a framework for analysis can act as a useful guide to an iterative design approach (Dillon 1994). So why can a framework be put forward when guidelines cannot? The case for a framework rests on its adaptability to actual design situations. It seems that Dillon's main misgivings in the realm of design guidance are reserved for what in another paper he calls 'formal guidelines' (Dillon 1989) - the more firmly prescriptive rules and guidelines that allow no room for the quirks of real-life situations and users. This roughly corresponds with Shneiderman's (1992) guidelines level, but does not accord with Smith's (1988) perspective, where, as we have seen, guidelines are expected to be translated in accordance with the needs of the particular design situation into rules or algorithms. However, in a review of a collection of guidelines for screen design, Dillon (1990) also raises several other valid points which do not hinge on matters of definition and interpretation. For instance, one frequently knows little about the provenance of the various forms of design guidance. Where have they come from? On what research or experience are they based? This is often unclear, although the more credible sets of guidance, such as Brown (1988) are as fully referenced as possible. Smith (1988) points out that (in user interface design at least) design guidelines and standards are mostly based on 'expert judgement and accumulated practical experience' (Smith p 884), as opposed to experimental data and quantitative performance measures. Judgement is the critical element - it is involved in the initial proposal of a guideline through to its selection and use by the designer. The designer must select the relevant advice, and judge how to weight that advice against other considerations where trade-offs must be made, and to convert the guidelines into appropriate rules for the application under construction. Finally, judgement must also be exercised in any evaluation of the successful use of those guidelines. Guidelines cannot be accepted as definitive in all situations. As Smith (1988) points out, codifications of guidelines based on judgement gives the impression that more is known and certain than is actually the case, and may undermine future necessary research. However, it is worth remembering that as Brown (1988) says, user interface design decisions need to be made even when there is no relevant experimental data, or in Smith's (1988) words, '. . .today's design decisions must be made today.' (p884). In any case there is considerable force behind the argument that it is not possible to generate comprehensive and useful design advice from experimental data, as discussed in Chapter 3. The number of variables involved is huge (Landauer 1991), and the experimental design problems similarly huge. To further compound the problem, interactive systems are a moving target - the pace of change is such that in-depth studies are likely to be superseded by the time results are sufficiently generally available.

One feature of these criticisms that should be noted is that just as there is a tendency to be somewhat vague about terminology, so there is a corresponding vagueness in discussion of faults associated with different kinds of design guidance. It is often not clear which particular criticisms are levelled at what kinds of design advice. Here Smith's approach is exemplary (1988) in that after clearly defining the kinds of design guidance he is considering, he goes on to consider the relative merits of these various forms. Standards are seen as not being particularly helpful to the designer, although their economic and commercial value is acknowledged. To those that argue that standards do not have to be so unhelpful, by incorporating elements of explanation and example, he responds that such 'standards' would more properly be called 'guidelines'. Neither standards or guidelines are specific enough to be directly applied. Guidelines should be useful in focusing attention on user interface design issues and causing the adoption of agreed design objectives. Rules can give clear practical inputs to the design process and ensure interface design consistency, but fail to provide guidance in situations where the rules don't fit. Design algorithms, although not yet common (at the time Smith wrote) are a way of speeding up the design process and further ensuring consistency. However, they are inherently inflexible and make exceptions and innovations difficult.

Despite these and other similar criticisms, other authorities have no problem in recommending a range of principles and guidelines. The HCI domain shows the greatest willingness to adopt such advice. (This is perhaps in line with the acceptance in the 80s of theory-based work, to which somewhat similar objections may be raised - see the discussion of the role of theory below). In the hypertext world some attempt has been made to produce similar collections, but to a lesser extent (Hardman & Sharratt 1990). The same goes for information systems development in general, and hypermedia systems in particular. The essential problem is that whereas flexibility is a desirable quality, the greater the flexibility, the less useful is the guidance.

One point that should be remembered is the distinction between feature-oriented and procedure-oriented guidelines. Whereas it may be arguable that feature-oriented guidelines are subject to Dillon's criticisms above, procedural guidelines are more concerned with the designer's task, and such criticisms therefore are not relevant.

4.1.2 SUMMARY: DESIGN GUIDANCE AND DESIGN METHODOLOGY

4.1.2.1 Is there still a case for a hypertext design methodology?

Criticisms of the various forms of design guidance are extensive. This being the case, what point is there in collating and representing this material for the designer? The view expressed here is that design advice can still be of value to the practitioner, although anyone using any kind of design advice should always remember the criticisms of such advice noted above and follow Smith's advice to exercise judgement. But the expansion of the PC, and hypertext and multimedia applications has resulted in what might be described as a democratisation of electronic document design. Many more people are seeking to use these systems to design documents. In many cases they attempt to transfer expertise from other areas of experience, with mixed results. Essentially design guidance provides a cheap and available substitute for the presence of expert training and supervision, something that is not readily available to many people. Many of the arguments put forward against the use of design guidance may be advanced against the direct advice of experts, and even one's own accumulated experience, which itself tends to be composed at least in part of a set of heuristics and principles. To return to the world of analogy, a coherent methodology would be valuable in the way that a recipe is valuable for the cook, providing a systematic approach and a stimulus for ideas. The recipe may not work in your kitchen with your ingredients and techniques, but it might be readily adapted, and if not it will provide ideas and techniques for next time. For the expert, a checklist of points to consider can always be useful, although this may come to have more of an evaluatory role.

With these points and issues in mind, the taxonomy was originally conceived with the aim of providing data about interactive system development relevant to hypertext which could be incorporated into a hypertext development methodology. The reasoning was that the various elements of design guidance - rules, principles guidelines, methodologies - that have been proposed over the years in relevant fields could be aggregated and classified, and where necessary condensed, to form part of the proposed methodology. The process of structuring the material into a taxonomy would also, it was hoped, provide some further insight into the nature of the hypertext design process.

4.1.2.2 Other contributions of the taxonomy

Although the main aim of this examination of design guidance was to provide data for the design methodology, it has other useful functions for this study. It has informed the work described in Chapter 5 on the design of a prototype hypertext electronic prospectus, providing guidance and direction at various points. It has also been instrumental in providing a better understanding of design procedures, which contributes to the construction of the design methodology, and proved useful in designing the case study methodology.

4.2 Methods and procedure:

4.2.1 METHOD

4.2.1.1 Background

The initial idea for this part of the project stemmed from earlier experiences in trying to build a hypertext system in which Shneiderman's (1989) 'Golden Rules' for hypertext were used. This suggested to the author firstly that design guidance could be useful, and secondly that hypertext was sufficiently new and strange as to warrant rather more guidance. An initial inspiration for this work was Hardman and Sharratt's (1990) analysis of two existing sets of human-computer interaction principles and guidelines for hypertext. This used two commonly cited sets of guidelines and principles for human-computer interaction, and extracted those thought relevant to hypertext development using a four-stage process. These were classified firstly as either principles or guidelines, and then according to which functional hypertext area they applied.

The present study is intended to be more comprehensive in scope, with a complete range of sources of guidance for hypertext design, including methodologies and models having been considered. On collection, the various sources of guidance were analysed and classified according to relevant characteristics, and then a synthesis of these was produced.

4.2.1.2 Methodological Issues:

This part of the research is an exercise in secondary research, a re-analysis of work by other researchers and practitioners (Williams et al. 1988). This has the advantages of enabling a wider view of the hypertext design situation, not subject to the intervention of this researcher in the initial data gathering. However, it is not always possible to detect weaknesses in the data as originally gathered, analysed and presented. It is also subject to the filtering imposed by the practitioner of secondary research.

To put this into context, the general approach of this research, the construction of a design methodology, can be seen as scientific project, a contribution to theory in the sense of an analysis and synthesis: analysis in the Cartesian sense, a breaking down into constituent parts of the various aspects of the hypertext design phenomenon, and a synthesis as in a reassembling of these parts into a complete methodology (Coyne 1995). The analysis would consist here of the breaking down of the topic into its constituent parts in the form of the creation of a taxonomy. A taxonomy is a system of classification based on consistent principles. In the biological realm it consists of the allocation of species to groups and sub-groups on the basis of similarity and difference. Its general value is that it encourages and compels consistency and coherence, and enables workers in a subject area to be sure that in using terms they are talking about the same thing. This enables what Shneiderman refers to as 'useful comparison' (Shneiderman 1992 p54). What is the point of taxonomy in this context? Essentially it adds shape and analytic structure to an amorphous collection of data. Early inspection of some of the design advice documents showed that consistency and coherency are qualities that are clearly lacking in a collection of design advice from different authors writing in different contexts and domains. A single system of classification was essential to reduce confusion. So the aim was to create a set of classes or categories which could be applied with some consistency across the various items of design and development advice encountered in the literature.

4.2.2 PROCEDURE

This section of the work consisted of five more or less identifiably separate tasks. As will be seen there were points at which the distinction between these tasks was blurred, but it is helpful, at least initially, to see the work in terms of these tasks. The first task was to evolve parameters for the selection of the source material. The second was to identify, assemble and list all the various items. The third task was the devising of appropriate categories to which the material could be assigned. The fourth task was to collate and allot relevant material to these categories by means of creating first sets of tables and then, in the case of the principles and guidelines, a database. The fifth stage was to sort the categorised material and produce an edited synthesis.

These stages are examined individually below, with the exception of the third and fourth stages, which in practical terms tended to overlap.

4.2.2.1 Evolution of parameters for selection of sources

The first task was to decide on parameters for the selection of sources. There are three issues here: what kinds of material should be included, what quality of material, and how much material should be included?

4.2.2.1.1 Choice of material

As the discussion above shows, many sources in the hypertext and multimedia fields contain design guidance of some kind. Decisions had to be made about what material should be included or omitted. In the first instance, it was decided that within the confines of this particular piece of research that no attempt would be made to collate all the assorted recommendations, tips, suggestions and ideas that appear in so many books and journal articles. Instead an initial decision was made to restrict this work to more formally presented sets of design guidance. These have various forms, and are called by various names as has been discussed above. They were first identified largely by their use of these names. This is not entirely fool-proof, however. Not all principles are of the design guidance type, nor all models or methods. Simply searching using such terms does not provide an exclusive set of suitable material. Judgement had to be exercised, and a number of sources were checked and rejected.

The second issue here was that of disciplinary parameters. What disciplines could offer relevant advice? The approach taken here was an eclectic one, on the basis that the design of hypertext in its broadest manifestation is a multi-disciplinary task that utilises skills from various disciplines. There is a body of guidance specifically for hypertext, and most of this was included. The main problem here was that some of this guidance relates to particular applications of hypertext, such as technical writing and CAL. Was this material relevant to the design of electronic prospectuses? The response here was to play safe and be inclusive rather than exclusive, at least as far as initial data gathering was concerned. Stricter parameters could be imposed later after more detailed consideration of the material.

HCI and, more broadly, human factors sources abound. Much of the material they contain is not directly relevant to hypertext design, being concerned with broader ergonomic issues. Two of the most widely cited and used set of HCI guidelines, those of Smith & Mosier (1986), and Brown (1988), were used by Hardman & Sharratt (1990) as the basis of a set of user interface guidelines for hypertext, and the latter's formulation of these was chosen rather than the originals (after cross-checking with the originals). It was decided not to use further sets of user interface guidelines although there are a number of others. The material gathered so far contained little information on the use of colour, so this aspect was supported by the inclusion of a set of principles and guidelines for the use of colour in interactive displays. The few hypertext development methodologies were identified and included, along with a group of descriptions of the authoring process for hypertext.

Architectural models of hypertext were included because although they are really intended to assist the developer, of hypertext systems, they can be used to offer guiding principles for the design of documents. A design method for desk-top publishing of conventional prospectuses was also included in recognition of the focus on electronic prospectuses. The sources are described in more detail below. The actual assembly procedure varied. Some of the sources included were already known to the author; others were located via references in the literature, database searches (OCLC, SSCI, Psychlit) and Internet searches.

4.2.2.1.2 *Quality of the material*

An area of concern is the accuracy and reliability of the status of the source material: its basis in research or practice, its validity, the degree to which it is generalisable, its history of use, whether the material has been evaluated in practice. We recall Dillon's (1990) observation that one of the problems with guidelines is that they may be contentious or totally wrong. The simple fact of publication does not render them accurate. This applies to all the material to be included, not just guidelines. The literature is extraordinarily varied in its approach to referencing and acknowledgement of sources, with many claims being unsupported and evidence unsourced. Models and frameworks are often similarly alchemic in their genesis, one exception being Dillon's framework (1994). As far as design methodologies are concerned, there is often little more than a reference to the author's own pedigree as researcher or practitioner. This is understandable where writers are attempting to synthesise a large range of ideas into a series of recommendations. There are also commercial considerations involved in the development of many ISDMs, which are often expensive products.

The various materials included are typically drawn from a wide variety of sources. One example, Brown's guidelines for HCI (Brown, 1988) drew upon experimental evidence, predictions from theory of human performance, principles drawn from cognitive psychology, principles from ergonomics, and evidence based on 'engineering experience'. As Brown (1988) notes, experimental data is in short supply. Given the scope and resources of this study, it was not considered possible to perform any critical evaluation of the sources beyond examining them for internal coherence and to utilise guidance sources which appear credible in terms of academic reputation of the authors or are commonly cited.

The age of the material was also a cause for concern. Old material may have been superseded, whilst new material may be untried. Where reservations or criticisms of particular sets of advice existed, these are discussed below, but are explored in more detail in the notes to the tables. But dealing with such material inevitably becomes, as Smith said (above) a matter for the exercise of judgement. In the event, material dating back to 1986 was included alongside more recent material. In one case (Horton 1990), material originally collected was superseded by a later edition (1994) and had to be updated.

4.2.2.1.3 *Quantity of material*

This was to a large extent determined by availability, this in turn being partly dependent on the parameters above. In practice there was less material meeting the criteria than perhaps was expected. One point to be considered was the appearance of new guidance. Material was collected over a period of two years, with new items appearing intermittently. (Interestingly, the tendency was away from guidelines towards more complete methodologies.) It was decided that there was no point in imposing an artificial cut-off point for the inclusion of new material; however, there was clearly a point beyond which it would be impractical to consider such material. The rule of thumb which was applied was to stop considering material when it appeared to contain no new contributions to the existing body of material.

4.2.2.2 Identification, assembly and listing of material:

In the light of these criteria, a set of published material covering principles, guidelines and design advice relevant to hypertext was assembled. These are described in brief below. More information may be found in the guidelines tables themselves (Appendix 1)

4.2.2.2.1 The sources

The following is a short introduction to each of the sources used, giving an explanation as to why each was chosen, and a short commentary.

Models and frameworks

Hypertext models are also discussed above. Examples of both architectural and process models are included. Process models are considered with methods and methodologies. Most of these have already been discussed in earlier chapters.

Dexter Reference Model (Halasz and Schwartz (1990)

This is perhaps the most widely known and used, hence its inclusion. This was originally intended to fulfil the first of the purposes mentioned above, to act as a reference standard against which a range of hypertext systems could be compared. However, it has also been used as a basis for the design of new hypermedia systems, for instance by Hardman (1994).

Hypertext Development Model (HDM) (Garzotto, Paolini & Schwabe)

HDM is a model of the hypertext development process characterised by the adaptation of techniques used in software engineering and database design.

Trellis (Furuta & Stotts, 1990)

A model of hypertext architecture.

HAM (Campbell and Goodman 1988)

Included as being a simple model of hypertext architecture which also acts as a reference for the more complicated models.

Design methodologies and authoring procedures

Several attempts at producing hypertext-specific development methodologies are included. A design methodology is a term most commonly found in information systems. It refers to a collection of procedures, techniques and other aids to design which, although something of a misuse of the word 'methodology', is commonly used. Methodologies generally consist of sets of phases or steps, themselves consisting of sub-phases, which guide the sequence of development and the choice of the techniques.

'Authoring procedures' requires a little explanation: it refers here to the kinds of step-wise lists of authoring tasks which are found in the hypertext literature, Shneiderman's being a fairly typical example. These are less detailed than the design methodologies, but provide an alternative in perspective to the database oriented perspective of the hypertext methodologies.

Asynchronous design/evaluation method (Perlman 1989)

This is a hypertext research oriented methodology based on data modelling. It is intended to aid hypertext design for research purposes, so it is asynchronous to allow for the continuous iterative refinement of the research process.

Van Vliet and Wilson (1993)

A database-oriented design methodology for hypertext.

Relationship Management Methodology (RMM) (Isakovitz , Stohr & Balasubramanian 1995)

Based on data modelling techniques, specifically entity-relationship modelling, but is more detailed and closely thought out than Van Vliet and Wilson.

Object-oriented Hypermedia Design Method (Schwabe & Rossi 1995)

This is based on HDM. It describes a four step process towards building hypermedia applications, beginning with domain analysis and proceeding through navigational design and abstract interface design to final implementation

General design methodologies & ISDMs: the waterfall model (Lee 1979)

A simple general model of the design process, analysis, synthesis and evaluation, was used to provide a contextual framework. This was developed in information systems design into the waterfall model (Lee 1979). This traditional model of the software design life-cycle was included as a further reference.

Shneiderman (1989)

A general account of the hypertext design process.

Martin (1990)

An account of tasks in the hypertext design process, concentrating on the early tasks such as information gathering.

Waterworth (1992)

An account of how to author hypertext from the bottom-up.

Principles, rules and guidelines

Horton (1990, 1994)

Horton's contribution comes in two parts. The first part consists of a fairly constrained set of guidelines from his book on on-line documentation. By 1994 this had grown to a largely hypermedia-oriented book, one of the better 'how to' books on hypertext, although with the focus still on online documentation it is quite specialised. The guidelines included in the database are selected points from the book and by no means account for all the design guidance contained within it.

Martin (1990)

A quirky collection of guidelines included with the process outlined above.

Balasubramanian (1994)

A set of guidelines distributed on the WWW.

Hardman & Sharratt (1990)

This was described above, an attempt to codify a set of principles and guidelines for user-centred hypertext design. They were based on Smith & Mosier, and C. Marlin Brown (1988). The original sources have been examined to check reliability, but these sources have not been used directly. The contents are almost exclusively feature-oriented.

Gould (1988).

This is interesting in that it combines a set of principles and checklists with a chronological framework for usability design. The advice is based on extensive experience and the HCI/usability literature. Its emphasis is procedural, focusing on practical routines for improving usability, and it is included particularly because of that emphasis, as opposed to the feature orientation of other HCI guidelines.

Schneiderman (1989), Kreitzberg (1989).

Two versions of a set of broadly similar principles found in a number of places.

Landow (1989).

A hypermedia-specific set of rhetorical rules.

Apple: The HyperCard Stack Design Guidelines (1989).

Produced by Apple, are included, despite their package-specific approach, as they form one of the most detailed sets of hypertext-oriented design guidance, including much that translates well to documents produced by other applications. Their principle disadvantage is that being HyperCard-specific they address a wider range of applications than the hypertext domain.

Brooks (1993)

A set of advice from the technical writing world is included.

Travis (1991): Colour display guidelines

These cover all aspects of cathode ray tube (CRT) display, and only elements which relate closely to hypertext design issues are included.

Thüring, Hanneman & Haake (1995)

Design principles and guidelines for hypertext based on a set of cognitive issues.

4.2.2.3 Developing and applying the classification

This section is an examination of the process of categorisation and the application of the categories, including details of the categories chosen and the reasons for their selection. Although logically consisting of two stages, in practice this work was an iterative process, with the production of categories being followed by their inclusion in draft tables followed by the inputting of design guidance. This would frequently result in amendments to the original categories and then to the tables as categories were seen to be unusable or irrelevant, or the need for new categories was revealed by closer acquaintance with the material.

4.2.2.3.1 Initial classification

The first step in classifying the data was to establish a set of categories and further sub-categories, if any. This raised several questions. What constituted a category? How should categories be generated and selected? How should categories be defined? The first question though is how many divisions there should be, and how large or small should these divisions be? Following from this, the question is, should these be subdivided, and if so, how, and to what extent? There is no point in imposing an arbitrary structure: the material itself should determine this. In response to the first question, how many categories should there be, three broad categories were selected. This accords with the structure used earlier in discussion of the various guidance types, and can be seen as three distinct kinds of guidance: 1) theories, models and frameworks, 2) design methodologies and authoring strategies, and 3) principles, rules and guidelines. The rationale behind this differentiation has already been discussed, along with some of the difficulties presented by differences in terminology. Its main attraction at this stage was that differences between these three broad categories are reasonably clear-cut. Another basic differentiation can also be seen, between design guidance that pertains to the product, its functionality, its user interaction style, and that which pertains to the design process. It is a matter for judgement as to which should form the initial categorisation; in this case it appeared simpler to allot elements to the three main categories first and then into the system or design process categories, as this distinction applies most usefully to the principles rules and guidelines category. The distinction appears in the taxonomy as a division between domain (feature-oriented) and scope (procedure-oriented) guidance, the terms being borrowed from Avison and Fitzgerald's (1988) classification for ISDMs, and the original distinction being borrowed from Gould (1988).

It was also clear that different approaches and modes for presentation would be required for guidance in each of these categories, a reflection of their qualitative difference. Having said that, it should be remembered that in practice there are overlaps between these categories. For instance, a design methodology will usually be based on a model of some kind, even though this is not immediately apparent, and design methodologies may also contain guidance of the principle/guideline type.

4.2.2.3.2 *Further categorisation*

Having adopted these categories, the second question is how might the various elements be further classified? The need for some differentiations within the categories was apparent from the material itself in some cases. For instance, the division of the 'guidelines' category into sub-divisions of rules, principles and guidelines suggested itself. Having distinguished a feature (domain) category, it was then logical to produce sub-categories which identified which features were addressed by the guidance item.

An initial list of possible sub-categories was compiled, with the emphasis on quantity rather than quality - that is, as many categories as possible were suggested without regard for their final value in the taxonomy. Reflection and discussion with colleagues, suggested a number of sub-categories, and the literature suggested more. This produced a number of items which were scrutinised for their usefulness and relevance and this in turn resulted in a number of sub-categories which are discussed below. Before examining the categories and their defining characteristics, a few general points regarding the problems involved can be made.

Hierarchical issues

Hierarchies pose a particular problem. It is easy, almost automatic, to equate terms such as 'principle' and 'guideline' with some notion of hierarchy - a principle is assumed to be of a higher level than a guideline, an overarching concept that in turn underpins the smaller scale guidelines. This is not necessarily the case. What is meant by 'higher level' in this context? Level of abstraction or breadth of scope? A principle may be abstract and somewhat general in its expression, yet be quite specific in terms of its scope in terms of stages of system development. Hence a hierarchical classification can be difficult to establish, and possibly not particularly useful in many areas.

Design methodologies and authoring procedures

As this sub-title implies, there are two obvious sub-categories here. Whether one considers hypertext system design in terms of methodologies or authoring strategies says much about one's preconceptions about the design process - do we write hypertext or do we design it?

The analysis and comparison of design methodologies is a subject in itself, and has been explored by Avison & Fitzgerald (1988), Jayaratna (1994), and others. Authoring procedures are somewhat neglected by comparison, although the literature review noted the work of Sharples, Wright and others on related issues. The two approaches represent different perspectives on the same problem, and it makes sense to treat them as sub-categories.

Design methodologies can be classified according to a number of parameters. Many of these have been the subject of study as part of the search for methods of comparison for ISDMs. There appear to have been no efforts as yet to develop any kind of comparative framework specifically for the study of hypertext design methodologies. The task of comparison between design methodologies is fraught with difficulties, and a number of methods for comparison have been suggested (Olle 1988, Avison & Fitzgerald 1988, Jayaratna 1994). These have even been subjected to their own comparison analysis (Savolainen 1991)

Avison and Fitzgerald's own framework was constructed in the light of other earlier attempts at comparison, and therefore provided a useful starting point, although clearly its concern with ISDMs as a whole would mean it lacked hypertext-specific concepts. The framework is composed of seven basic elements, some of which are further sub-divided. The original framework is shown below (Table 4.1), with notes as to the usefulness for categorisation of each element:

Philosophy	Seen as highly important by Avison and Fitzgerald.
Paradigm	For instance science versus systems, realist versus nominalist. Worth noticing that the methodology versus authoring split is of this kind, and many of Avison and Fitzgerald's criteria apply to the latter also.
Objective	This refers to the stated objective of the methodology. The kinds of distinction Avison and Fitzgerald are interested in are those such as between methodologies that are only for computerised information systems and those that are looking for the most appropriate solution, regardless of type.
Domain	Refers to the areas of application which the methodology purports to cover - a particular problem area, an organisation? The main difference here would be between hypertext-specific methodology and the prospectus design method included - which already falls into the authoring strategy category.
Target	This refers to the applicability of the methodology to particular problems or organisations.
Model	Based on the idea that every methodology has an underlying model of reality, a way of representing the salient facts about the system or situation. Relevant, although not necessarily using the kinds of model aspects mentioned by Avison and Fitzgerald.
Techniques and tools	Most of the methodologies concerned do not elaborate on these aspects, not being commercial methodologies.

(Table 4.1) The original methodology framework (based on Avison and Fitzgerald 1988, p286).

Scope	This concept covers the stages in the system development life-cycle that the methodology covers. Useful, in so far as not all methodologies pay equal attention to all parts of the systems development life-cycle - but the problem here is that description in terms of the systems development life-cycle implies some agreement as to what constitutes the steps of that life-cycle - which is not apparent - and that the life-cycle model is applicable at all, which proponents of iterative methods suggest is not the case.
Outputs	This refers to the end-product of the methodology, such as a specification or a prototype. Perhaps useful.
Practice	Covers background, user base and participants.
Background	Refers to the context in which the methodology was developed, for instance in industry or the academic world.
User base	Very hard to ascertain in the current context
Players	Ditto
Product	What do you actually get? Again, not particularly useful - all the methodologies examined simply provide an account of what needs to be done - there are no tools, books, courses associated with them (yet). Gives no differentiation - therefore not useful.

(Table 4.1 continued) The original methodology framework (based on Avison and Fitzgerald 1988, p286)

From these, the following concepts were identified as of potential relevance and use:

- Paradigm
- Domain
- Target
- Model
- Scope
- Outputs

However, this was felt to be unsatisfactory in two respects. Firstly, this is not enough to identify differences between the hypertext methodologies and strategies. Many significant differences would be ignored. What is needed are some hypertext-specific criteria. Going back to the discussion about hypertext definitions in Chapter 2, it will be recalled that regardless of arguments over definition, it is clear that hypertext is characterised by some combination of links, nodes and a navigation or traversal structure. It makes sense therefore to look for differences in the ways in which these were handled by the various design methodologies. At this point it was decided to examine the methodologies themselves more carefully and to see what differences became apparent. This was done by drawing up a set of tables which used a loose framework, derived from conventional accounts of the design process, of analysis, synthesis and evaluation, and then entering a selection of the methodologies. This suggested that a key difference lay in the extent to which the different methods used formal methods of modelling the hypertext structure. This could form a set of sub-categories in the models category. Secondly, it was apparent that the general design approach, whether prototyping, waterfall or function-based (Savoleinen 1991) was not identified by these criteria. A number of further key criteria were identified by reference to other comparison frameworks (Savoleinen 1991, Jayaratna 1994), which were included in a revised framework. Sub-divisions for the various categories were included where these could be anticipated. This gave the framework shown below:

Philosophy	–
Paradigm	The principal relevant paradigms seemed to be the database design paradigm, the object-oriented paradigm, and the document authoring paradigm.
Objective	–
Domain	–
Target	Maybe relevant, but most material obtained appears to be aimed at any kind of hypertext design problem
Role of models	abstract model? descriptive model? specification model?

(Table 4.2) Revised framework

Techniques and tools	Not always possible to identify
Scope	Some variation between methodologies, usually in the emphasis on analysis and evaluation stages
General design approach	Prototyping, waterfall or function-based
Outputs	Specification, prototype, final implementation?
Practice	–
Background	Principal relevant distinction between academia and commercial/industrial
Users	–
Participants	–
Product	What does the methodology actually consist of?

(Table 4.2 continued) Revised framework

This was considered an appropriate initial classification for the data, but it was expected, and subsequently proved to be the case, that application of the categories would generate modifications. The main problem that arose was that it proved very difficult in practice to separate out the notions of domain and target as used by Avison and Fitzgerald. Their descriptions of these concepts does not allow for clear distinction between the two, and it was decided to simply retain one 'domain' category. So the final representation of the framework looked like the completed example overleaf (Figure 4.3):

Methodology:	Relationship Management Methodology (RMM)
Source:	Isakowitz et al. (1995)
Reference No:	001/b
Philosophy	
Paradigm	From the science paradigm - formal methods
Stated objective	To provide a methodology for hypertext applications where there is a regular structure to the subject area which can be seen in terms of classes of objects, definable relationships between classes, and multiple instances of objects within each class. This means artefacts such as product catalogues, hypermedia front-ends to databases etc.. Claimed to be especially suitable for applications that need frequent up-dating.
Domain	Fairly narrow - assumes that feasibility, hardware choice & requirements analysis have been done, and that a hypermedia solution is necessary.
Target	As stated.
Role of models	Uses a data model to describe the information objects and navigational mechanisms. This, called Relationship Management Data Model (RMDM) has its ancestry in database modelling, via Garzotto, Paolini & Schwabe's HDM & HDM2 data models. The model is crucial - effectively this methodology is composed of the model core, plus a more detailed set of instructions and design steps.
Techniques and tools	No specific techniques and tools at the time the source article was written, although a set of support tools was under development (RMCase - Diaz & Isakowitz 1995). Includes extra guidelines.
Scope	Main concern is with stages from initial design to testing and evaluation, ie does not include feasibility, hardware selection and task/information requirements specification.
General design approach	Broadly a step-wise process, but with iterative elements, and some awareness of the propensity of designers not to restrict themselves to stages.
Outputs	Final implementation.

(Table 4.3) An example of the final design methodology framework in use

Practice	No information
Background	Academic
Users	Not known
Participants	The development team
Product	Not a commercial product - at time of writing, only the articles existed.
Comments	

(Table 4.3 continued) An example of the final design methodology framework in use

Theories, models, frameworks

In practice, for the purposes of this work, this category concerns models and frameworks. The main distinction here was seen as being between models that describe system features or architecture, those which model the design process, and those which model various aspects of the user (those for which Dillon (1994) prefers the term 'framework'). This is reminiscent of Gould's distinction above between feature-oriented and procedurally- or methodologically oriented guidelines.

Guidelines and principles

Gould's (1988) division of guidelines etc. into two kinds was the first sub-division here. The procedurally or methodologically oriented guidelines, such as guidelines about user testing were again labelled 'scope', and those which refer to features which are considered desirable labelled 'domain', the terms being borrowed from Avison and Fitzgerald (1988). This provided a useful broad division between the part of the development process to which the guidelines apply as opposed to the functional area. A sub-division of the 'domain' material was possible based on the common approach of classification via the functional area targeted by the guideline. This is useful in the more focused levels of the guidelines category, for instance Brown's (1988) HCI design guidelines are categorised on a functional basis that works well at the lower levels, but does not apply at the more general system levels, where general principles are outlined on an individual basis. Hardman & Sharratt (1990) also use a set of functional categories.

<u>The guidelines sub-categories in detail</u>

The guidelines and principles varied greatly in a number of ways. These various qualities or attributes are explored below and their use as categories and sub-categories in this context are discussed. As mentioned above, the approach taken was to generate sub-categories from qualities inherent in the various advice items rather than imposed from external classifications of any kind. This meant an initial assessment of all individual items in order to identify and list relevant attributes. In practice this meant listing all the guidelines first, then examining them. There are substantial differences between guidelines identified as procedural and feature-oriented, so these are dealt with separately.

Some attributes were examined but were not included because although of interest they were either very difficult to conceptualise and apply or they contributed little value.

<u>The categories</u>

Each guideline was given the following categories, some of which are further divided into sub-categories:

• *Number*

Useful to give each item and component a unique identifier.

• *Item & Components*

In early tables, the guidance elements were listed as items with subsidiary components when necessary.

• *Sources*

Obviously attribution is essential as it was envisaged that a composite collection of guidelines would emerge. Authorship at component level had to be clearly identifiable.

• *Source subject background*

As noted above, design guidance with a relevance to hypertext and multimedia can come from a variety of subject areas and disciplines. Because of the intention to mix guidelines, it was felt that subject origin should be identified for each guideline to enable users to evaluate the guideline effectively.

• *Type according to author*

The authors of all the sets of advice in this table give some indication as to the area at which their advice is aimed. In some cases this is quite specific; in others it may be considered generic. It is not always the same as the subject background. Again, this attribute is not particularly edifying, but does give some idea of the author's intentions regarding the component.

• *Importance*

It was felt that some way of indicating the significance of each item of design guidance. It is clear that various authors have different opinions as to the importance of their advice. However, this quality is not always overtly indicated by the original authors. Although the use of such terms as 'rule' may be indicative, this is not necessarily the case. To what extent should a guideline, for instance, be followed? Some authors do explore this issue but many do not. In the end, it was decided to omit this item on the grounds that it was likely to prove excessively subjective.

• *Level*

This was troublesome. It is conceivable to see principles and guidelines as proceeding in levels from specific detailed advice and rules to high level general principles (shading into large-scale theory), and it was initially considered as significant to note these distinctions. The level of advice can vary from strategic/high-level to keystroke/mouse level, and Schneiderman (1992) as noted previously, distinguished four levels of guidance for designers of interactive systems: 1) high level theories or models; 2) middle-level principles, 3) specific and practical guidelines and 4) strategies for testing. As only 3) and 4) are considered in this category, this is not as useful as it might have been. However, this simple dichotomy does appear to have some value and was therefore retained. In practice though it proved unexpectedly difficult to distinguish between guidelines and principles.

• *Notes*

Self-explanatory

• *Feature domain/Domain of application*

This term was adopted to refer to the feature guidelines. This could be further subdivided into a split between functional and non-functional aspects, as in software engineering, but it was felt that there are some difficulties in separating such aspects with hypertext documents, and that therefore such a division would not be particularly useful. So this category subdivides into a number of feature areas such as information content and user interface. The feature areas were decided by making individual decisions about the subject of the various feature-oriented guidelines etc. while they were being entered. This generated a number of terms, some of which were vague and which were not consistently applied to start with. For instance, 'human-computer interface design', 'usability' and 'user interface' were all used to describe guidelines covering similar features at this stage. These terms were sorted and standardised, and a second pass made through the database to apply the terms more consistently. In the example described, this meant adopting the term 'user interface' as referring to a feature rather than 'usability' which describes an HCI concept, and 'human-computer interface design' which describes a design task and fits better into the procedure category.

• *Scope (design procedure stage)*

Scope is a term borrowed from Avison & Fitzgerald's (1988) discussion of comparison of ISDMs. The point is that some design guidance, rather than referring to document features, refers to aspects of the design process. For instance, guidelines about task analysis directly concern the early stages of the development process. This is not to deny that such items do not in some ways relate to document features, but their initial emphasis is on process and the design task, how to design, rather than on features and document quality, what should be designed. Given that this is the case, certain guidelines etc. can be seen as aimed at particular parts of the design process, and this category and its sub-categories are intended to isolate these.

The sub-categories here were again produced on an ad hoc basis during the process of guideline entry into the tables. They were then listed and compared to the design methodology material. This enabled the production of a number of sub-categories that could be related fairly easily to any of the methodology components. These were then reapplied to the guidelines. As with the domain sub-categories, there were contentious items, but it was felt more important to be consistent in classification than to strive for perfection, which meant in practice using relatively broad sub-categories.

Construction of first tables

The guidelines and principles were entered into a table (using a word processor). As indicated above this took place concurrently with the final development of categories.

Conversion to a database

The table format was useful to begin with, easy to modify and requiring the minimum of initial design. However, it was found to be severely limiting when it came to sorting the material, so it was exported to a database (FileMaker Pro). This simply adopted the various categories as fields. No distinctions were made in cases where categories had sub-categories, with data being entered in these cases into a field with a DOS-style path representation of the hierarchy. The items and their components were all given individual entries, but components were numbered to identify their relationship to items. The main problems here were the practical ones of maintaining the integrity of the data, which had to be carefully checked. The data were then sorted to provide a number of sub-sets based on chosen categories and sub-categories.

Sorts

In the first instance, feature-based guidelines and procedure-based guidelines were separated. The feature-based guidelines were then sorted by functional category, and the procedure-based ones by scope.

Final tasks

This was followed by the removal of all redundant (superfluous) examples, leaving the clearest expression of each guideline. Classified sets of guidelines for different parts of the design process and for different features were then printed out (Appendix 1).

Collation & presentation of the information

The material was originally collated in tabular form. The first problem experienced was reminiscent of Avison and Fitzgerald's (1988) concerns about their scope concept: a tabular form easily leads to the imposition of a step oriented pattern which does not always adequately represent the material. The original tabular form was therefore modified to include more extensive footnotes and commentary. However, on entering some of the methodologies, it was found that this did not make for easy comparison, so the table was abandoned, and the template or pro forma shown above (Table 4.3) which utilised all the categories and sub-categories described earlier was used. The methodologies were all entered into this. Certain categories benefited from further diagrammatic representation, and these were produced as required. The standard categories enabled an adequate level of cross-comparison.

Collation & presentation of the information - models etc.

This was relatively straightforward in that most of these models use a layer metaphor, enabling the compiling of a set of diagrams using the same structure.

4.3. Results

The survey/taxonomy resulted in a thesaurus of relevant terms, a database of principles, guidelines, rules and other design advice, with categorised listings of these, and a collation of design methodologies. This was used to produce sets of guidelines relevant to specific stages and features in the hypertext document design process. It also provided input for the methodological level.

4.3.1 PROBLEMS

The general value of design guidance has already been discussed. How useful this particular taxonomy of design guidance is depends to a large degree on how well the taxonomy has been constructed. This can be considered in terms of the traditional scientific concerns of validity and reliability.

4.3.1.1 Validity

This study is not really concerned with measuring or with hypothesising, so it is perhaps hard to see how the concept of validity relates. Validity concerns the degree to which a study's measures and research design measure the phenomena or concepts that they purport to measure (Williams et al. 1988). The main aspect of validity that is relevant here is external validity, which is concerned with the extent to which a study's results are generalisable to other situations. What this means in this context is the degree to which the various design guidance elements can be applied to the design context of small to medium hypertexts exemplified by electronic prospectuses. A number of criticisms might be raised here. First, the various elements are originally designed to be relevant to a range of more or less specific areas, none of which precisely corresponds with the area of design concerned. The solution here has effectively been to take a wide-ranging approach: to collect sufficient material to obtain at least something of relevance. The danger here is that the relevant and useful material can be contaminated with the less relevant and useful guidance, whilst the prospective user is given no opportunity to distinguish material.

4.3.1.2 Internal validity

This normally concerns the extent to which the measures and research design operationalise the concept under study in a manner which precludes alternative explanations for the results (Williams et al. 1988). The aspect of this which is particularly relevant in this context is the possibility for biases. As one person's interpretation of an extensive body of material, this classification and analysis is open to accusations of subjectivity. This is unavoidable to some degree, and the principal protection against bias here has been the presentation of a full and open account of the process by which material was selected and graded. The best protection against bias is to have the process conducted by more than one person, but this is dependent on resources, and was not possible in the course of this project.

4.3.1.3 Reliability

Reliability concerns the extent to which measures can operationalise a concept consistently over time.

These considerations include some of the ambiguities and contradictions pointed out by other commentators on guidelines and reviewed above, but perhaps most crucially the process of individual selection and categorisation, followed by the use of the author's judgement in the process of synthesis, means that something less than perfect is inevitable.

4.4 Implications for a design methodology

There is no doubt that anyone considering the production of a design methodology for hypertext has to take into account previously produced design guidance. The question is, to what extent? The alternative design methodologies proposed for hypertext clearly have to be taken into account. Even in their inadequacies they provide a way to see more clearly what a hypertext design methodology requires. So the analysis and comparison of the alternatives produced here provides a basis and inspiration for the design methodology. An understanding of the role of hypertext models is also useful in further understanding the position and function of design methodologies with regard to hypertext.

The principles and guidelines have two functions in a design methodology, depending on whether they are concerned with features or with design procedures. The former are resources to be drawn on during hypertext design. The latter are elements which may integrate with the more general high level elements of design methodologies to provide a more detailed methodology than so far exists in the hypertext field.

How do these elements come together to form a design methodology? This process of construction is considered in detail in Chapter 7. First the other sources for the design methodology must be considered, beginning with the design and construction of an electronic prospectus.

5 Electronic Prospectus Design and Implementation

5.1 Introduction

This chapter describes the role in the development of the design methodology of the author's own work with a series of HyperCard stacks which were developed to familiarise the researcher with the subject area and task domain. This proved useful in sensitising the author to some of the problems, issues and decisions which are involved in hypertext design for documents like electronic prospectuses. More specifically, this activity led to a better understanding of the hypertext design task. A further benefit of designing a hypertext document was that it proved an appropriate context for examining and testing aspects of the design methodology. The HyperCard stacks were used as an exploratory tool, with a number of techniques being tried out and improved using the HyperCard electronic prospectus as a test-bed. A complete account of the electronic prospectus development is found in Appendix 2. The account below takes a critical look at the author's own attempts at prospectus design and examines some of the key issues raised by the development.

5.1.1 RELATIONSHIP TO OTHER PARTS OF THE RESEARCH

The electronic prospectus was originally seen as a way of learning about hypertext design for this domain of information provision by practice and reflection on that practice. A important element in this is the notion of preunderstanding.

5.1.1.1 Preunderstanding

This is a term used by Ödman to describe the previously acquired insights and understanding of the researcher in a project or organisational setting (Ödman, in Gummesson 1988, p53). These can obscure features of the situation under investigation because the individual's preunderstanding prevents a proper interpretation of events. Insufficient preunderstanding means the researcher must spend a disproportionate amount of time gaining basic background information about a situation or process. By having hands-on experience of the electronic prospectus design situation and process, the author was better positioned to understand and interpret the design tasks and issues experienced by the case study team.

A further aspect to preunderstanding is the notion that some kinds of knowledge are of a kind which is not communicable, as in knowledge of physical performance (Gummesson 1988). Where aspects of physical performance are concerned, as is the case in any design process, it is useful for the researcher to have some direct experience in order to enhance preunderstanding. Gummesson (1988) distinguishes two types of preunderstanding to incorporate this distinction: preunderstanding at first and second-hand.

This experience of design practice also had a direct contribution to the design methodology in allowing the author to consider the efficiency and effectiveness of a number of techniques such as diagrammatic representations of hypertext structure. Following from this, the electronic prospectus eventually came to be used as a test-bed for different aspects of the design methodology whilst these were being developed. It enabled testing of such elements as structure diagramming techniques in a realistic setting.

5.2 Method

As indicated in the chapter on methodology (3), this dissertation is essentially constructive research. This part of the research is concerned with exploring the hypertext document design situation. For this part of the research, the author designed a series of prototype hypertext electronic prospectuses, which provided information about the hypertext design task. This kind of approach reflects that found in other technological fields, and is common practice in hypertext design and in related fields such as information systems design.

5.2.1 METHODOLOGICAL ISSUES

This approach has the advantage of direct contact: as indicated above the author's experience gives direct knowledge in a way that no other method can. However, as scientific investigation, such direct experience is frequently criticised as having significant and severe limitations. These criticisms can be met, but they deserve serious consideration first. In particular direct experience can be considered to be subjective and prone to bias, and to be incapable of generalisation to a wider population. This is partly a matter of perspective. Such criticisms are based on the positivist tradition of scientific inquiry, which is concerned with developing general explanations on the basis of regularities and causal relationships in the phenomena under investigation.

An alternative paradigm exists, sometimes known as anti-positivism, although this presupposes an antagonism to positivism and the scientific method which all its proponents do not hold. This upholds the notion of validity for the viewpoint of participants, and rejects the notion that there can be impartial observers in most if not all investigatory situations. This perspective is also sometimes known as hermeneutics or interpretivism.

In the spirit of such research, the account below takes a reflectively critical stance in examining the author's work on the electronic prospectus.

5.3 The electronic prospectus design process

5.3.1 INTRODUCTION

A hypertext electronic prospectus was designed and constructed, initially based on the existing College prospectus. The information content was imported, edited and restructured manually into a hypertext structure. The account below concentrates on the issues and implications raised by the development. The design and development process is not described in detail here but a complete account of the process is found in Appendix 2.

5.3.2 DESIGN OF THE FIRST PROTOTYPE

5.3.2.1 Feasibility

Investigations into project feasibility covered IT provisions in Scottish schools and colleges, the resources available, the authoring approach and the software available for this, and an examination of the existing paper prospectus.

A number of hypertext software packages were investigated for production of the electronic prospectus document, using a list of products based on texts (Nielsen 1990) correspondence with colleagues, and reviews. Budgetary considerations at the time were critical in ruling out the most attractive prospects, Director and Guide. In particular, Director has attractive multimedia design features, and Guide has good facilities for working with existing text files and producing hypertext with a hierarchical structure. HyperCard was the eventual option. This was based partly on the financial constraints mentioned, on the author's previous experience with HyperCard, and awareness of the Apple bias in Scottish schools. Conversion to PC format could be done later using a conversion utility. Using HyperCard was subject to the constraints that have been documented elsewhere (eg. Kahn 1989), and it was not the preferred choice. However, this kind of limitation probably reflects the reality of software choice. Choosing the right software is critical, as the choice of software can greatly affect the design options open to the author. The approach used by the author could have been more systematic. For instance, Horton (1994) gives a ten-point procedure for software choice for on-line documentation which may help to systematise this process. However, this is rather over-elaborate for many circumstances, as it involves the acquisition of working copies of the best three to seven products, the construction of chapter-sized documents in each, and the construction of a complete book-sized document. In the end real constraints in terms of skills and costs are likely to be as significant.

5.3.2.2 Requirements analysis: stakeholder analysis, user analysis and task analysis

A formal requirements specification was not drawn up, but analyses were completed, including stakeholder analysis, user analysis and task analysis. This produced a number of documents which were used to guide later design.

5.3.2.2.1 *Stakeholder analysis*

The stakeholders are all those having an interest in the system under construction. This means not just the users but all the other parties concerned in support, and in the outcomes of use. The designer of any artefact has to balance the often conflicting requirements of all other groups concerned. This requires identification of these stakeholders. An analysis of the stakeholders and some of their possible requirements was made by identifying the various possible scenarios of use. This attempted to identify the possible stakeholders in the system and to discuss the scenarios within which the 'QMC electronic prospectus' might be used. It also suggested the tasks that each of the scenarios requires the system to perform. This analysis only discussed the above in general, as this stage was to precede a more specific user and task analysis. The purpose here was to identify the possible interaction scenarios with the stakeholders in an attempt to gain some insight into requirements for the electronic prospectus document. This technique is useful, but can be unsystematic in use, and depends very much on the imagination and contextual knowledge of the design team.

5.3.2.2.2 *User analysis*

The task and the user are integral to each other. What is done and who does it cannot be disconnected. However, it is helpful for understanding to make some separation between the two. This section examines issues surrounding the question of the user and the user's task in designing an electronic prospectus, and looks at the strengths and weaknesses of methods used.

are doubts as to whether such an approach is possible, just as user modelling is criticised. Landauer (1991) suggests that there are far too many variables to enable human cognition and behaviour to be modelled in any useful way. Dillon (1994) explores this more extensively, suggesting that despite the anti-theory feelings of some commentators, such as Landauer, the desired position is one in which good theory can support empirically based design. Dillon (1994) goes on to suggest that such a framework or model may be useful in the design of electronic text. This seems reasonable in that it is inevitable that there must be some degree of generalisation involved in task description if the task described is not to be tailored to specific individuals. A task analysis approach that leads to the development of an idealised normative task model has been developed by Johnson (1992). A commoner approach is the development of a task list often represented by a functional specification (Mullins and Treu 1993).

Dillon (1994) has explored the issue of task in hypertext, suggesting a 'how, why, what' approach to task analysis for texts and hypertext:

- why they are read;
- what type of information they contain;
- how they are read;

This would be difficult to apply at a detailed level, but it does offer a structure for task analysis for hypertext.

To date there is no task model specifically relevant to the designer of electronic prospectuses. Prospectuses represent an area of information seeking and use in which the information task is transient and particular to the individual user. The deeply personal task of finding out about institutions and courses, and then choosing appropriately is one for which there can be no task expert. Very few individuals go through this procedure more than a few times. Those who perform the initial information seeking task may do so more frequently in a professional capacity, but such professionals are unlikely to be directly involved in the decision-making part of be no appropriate user models in this area, so it is not possible to make any reasonable comment on their practical applicability for hypertext design.

In general, the problem with user analysis is that unless some existing work exists, such as in this case the work of Keen and Higgins (1990, 1992), a comprehensive user analysis is likely to be an inappropriately large task for anything but a major project with a large budget. Techniques such as the use of scenarios are useful devices, but do not provide empirical data.

5.3.2.2.3 *Task analysis*

Task analysis is a generic term for a number of techniques used originally in general training and more recently applied to interactive system design. A task can be defined as "... the mechanism by which changes are effected in a given domain." (Johnson, 1992, p155), or more narrowly as "the study of what an operator (or team of operators) is required to do, in terms of actions and/or cognitive processes, to achieve a system goal." (Kirwan and Ainsworth 1992), although recent commentators have noted the variety of tasks, and the dangers in taking too narrow a view of task (Draper 1993). Task analysis can however be a useful technique where the work or activity concerned is fairly constrained, as in this instance.

Task analysis can be more or less empirical in its approach. Early techniques for task analysis, which were originally directed at practical tasks, such as Hierarchical Task Analysis (HTA) (Annett and Duncan, 1967) are largely empirical, using techniques such as interviews, observation, and employing secondary sources such as manuals and other documentation. Such techniques have been considered as inadequate for dealing with cognitive tasks, although Shepherd (1989) argues that this is not the case. Cognitive task analysis attempts to take into account the knowledge that people bring to a task situation, and to analyse, model and eventually to predict that knowledge (Johnson 1992).

Task models

It would be useful if analyses of task in all situations could be rendered unnecessary, or at least minimal by the production of idealised models of the task situation. There

Existing knowledge

Where the user population is relatively significant, there is a good chance that there will be existing information about that population. The author utilised a number of existing sources. Much work on the potential student population already existed, in particular two surveys on the English population (Keen & Higgins 1990, 1992), and a Scottish telephone survey (Scottish Office Education Department 1993).

Using these sources was useful in that it gave a breadth that would never normally be available to the individual hypertext author with constrained resources. The scale of a project like electronic prospectus development does not warrant the use of extensive user surveys, and it is essential to investigate other sources. There are reasons for caution in the use of such material however. In particular, the surveys concerned were commissioned and designed with an entirely different purpose in mind. Hence although prospectus use is covered in a limited way, there is much material which is no more than interesting background. However, points such as the way that knowledge of the universities amongst young applicants proved greater than that of the polytechnics and Colleges is of great value. In terms of information content for the prospectus, this means a consistent emphasis on university-level status, both at the level of corporate identity, and at the level of information content for courses and departments.

Theoretical knowledge: user models

It has been argued (Card et al 1983) that it is both useful and possible to have a theory based science of the user that can enable initial design decisions to be taken without the use of empirical techniques. The usual forms that such assistance can take can be user-oriented or task-oriented. User-oriented forms are typically known as 'user models' or the like, as described above, whilst task oriented ones are 'task models' etc.

User models have in the past been suggested as a way of circumventing the need for detailed user knowledge in every instance of system development, for instance Rich (1981). However, there are as yet no workable models (Preece 1994) and serious doubts as to the value of user models exist in any case (Dillon 1994). There appear to the process although they may be indirectly involved in a supporting role. However, there are ways of making valid generalisations about the users' tasks in this context.

One possibility is Marchionini's (1992) model of the information-seeking process. This provides a useful way of conceptualising the task in the college choice situation. The task concerned in this instance can be characterised in the first instance as an information-seeking task. Marchionini (1992) describes the information-seeking task as a form of problem solving in which 'either or both the information sought (problem) and the search process (solution path) may be simple or complex.' This can be described in terms of five functions that form the information seeking process: defining the problem, articulating the problem, selecting the source, extracting the information and examining the results. The relationship between these is shown in the figure below

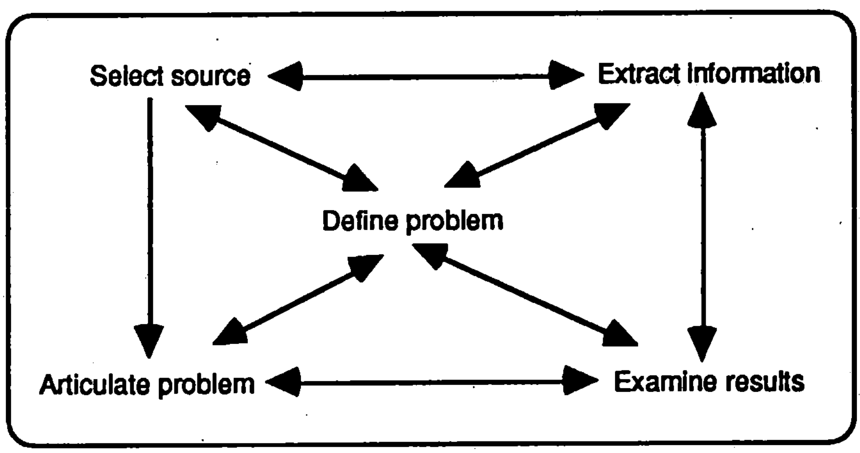


Figure 5.1 Information seeking functions (after Marchionini 1992)

These functions exist together in a non-linear relationship where any of the different functions may be engaged at different points in the search process. So for instance examination of results is not necessarily an end to the information seeking process but may lead to a redefinition of the problem or an extraction of new information. This corresponds with scenarios of the prospectus use task.

This is likely to be followed in many cases of prospectus use by a decision-making task, whether this be to select a specific college and course, or something less defined such as a decision not to go into higher education, or a decision as to a general area of study. This cannot be easily unravelled from the information-seeking task as the process involves the user constantly checking the information found against the decision-making task for relevance and usefulness.

McKerlie and Preece suggest a broad categorising of task goals for hypertext use into four types:

- finding an answer to a particular question (ie searching task)
- gaining a sense of scope for the information (ie browsing task),
- exploring a particular concept (eg learning), and
- collecting and tailoring information (eg. organising or synthesising)

(McKerlie and Preece 1993 p36)

Apart from the career professionals, users are relative novices, although they are likely to gather expertise in prospectus use a) through repeated concentrated use during the course selection period, and b) to a lesser extent, through following a series of courses at successively higher levels.

Task analysis conducted for this project

The first prototype involved a more or less direct transformation of the existing text prospectus. This formed the nucleus of the information to be included as it represents the institution's evolved view of what the information content should be. It was apparent from discussion with members of the Department and representatives from the PR department that no great deviation from this would be expected.

Empirical task analysis at this stage consisted of a series of pattern notes conducted with a range of participants including prospective students and new students. New students were chosen partly for their easy availability but also for their recent familiarity with the course choice process: they provide recent experience and the benefits of hindsight.

The pattern notes

Pattern noting is a task analysis technique suggested by Jonassen et al. (1989) for task analysis in hypertext. It belongs to a family of techniques known as concept mapping (Jonassen 1996, McAleese 1996). It works by asking the subject to produce a network of related concepts rather like that produced by mind-mapping (Buzan 198?, in Jonassen 19?), and is in fact based on Buzan's earlier work. Its value for hypertext is that it can produce a network structure rather like a hypertext net that can provide the basis of a hypertext structure. Apart from the relative ease with which it was said to present a network structure, this technique appeared attractive because it is relatively easy to administer and requires no special equipment apart from a pen and an adequately sized piece of paper.

The rationale here was that the better part of the information *content* was fairly well-known on the basis of many years experience with paper prospectuses and student recruitment in general (although higher education institutions are of course as capable as any others of making and holding unwarranted assumptions). This information to a greater or lesser extent has to be regarded as a compulsory element although the individual organisational context is important here. It is for instance impossible to imagine a useful prospectus without details of courses, entry requirements and admissions procedures. Although users are important, the needs of the sponsor must also be considered, especially when they are paying the bill. In many cases, users and sponsors are effectively on the same side: what is good for the user is good for the sponsor. This is not necessarily the case in a marketing context. The goals of the users and the goals of the sponsor can be very different. What *was* a relative unknown was how people saw the structure of that information, the relationships between the various topics.

A pattern noting approach was therefore used to elicit this kind of information. A range of students, including first year and prospective postgraduate students was asked to draw representations of the knowledge that they required to choose a course and a place of study, and the resulting networks were analysed. The students were not selected for representativeness but rather for their potential for generating a full range of required information. The first year students were chosen on the basis that having recently been through the experience of choosing a college and course, they had the experience still fresh in their minds, plus the benefits of hindsight. This approach reflects the logic used with such qualitative methods as focus groups (Morgan 1988). The procedure is repeated until the majority of items and connections being noted become redundant.

The factors included in the pattern notes were then used to make a set of diagrams displaying the relationships between the various elements as seen by the participants (D02, D03, Appendix 2c).

This approach to task analysis provided useful information, but poses certain problems. It does not give any indication as to physical aspects of the task. It does not give any indication of users' search strategies. It is also liable to produce somewhat ephemeral information (Jonassen 1996). What users represent on different occasions can change significantly.

5.3.2.3 Information content analysis

The existing prospectus was examined carefully, and a table of headings and sub-headings constructed (Table P03, Appendix 2b). Most of this material was already broken into chunks in the form of short paragraphs. Availability of existing prospectus text files and graphics was explored. There were no problems in obtaining prospectus files, but these were in PageMaker form, and required conversion back into ASCII text files.

It is hard to separate information content analysis from information content design. Analysing the information content in practice involves making preliminary decisions as to what information content should be included, and what the nature of that information is. But these preliminary decisions can also include such decisions as whether or not to retain the existing document structure, which are inherently information design decisions. However, some distinction is helpful to the designer.

5.3.2.4 Information design

The first information design decision was whether to retain the existing document structure or to develop a new one on the basis of the new knowledge gathered. In any case where there already exists a paper version of the hyperdocument proposed there are the two possibilities here: to convert the original text or to author from scratch to suit the hypertext medium. At this point an early prototype was constructed in order to assess the position. This was based on an old HyperCard shell from the Hypertext '89 Conference which was adapted. This had been developed in order to show that HyperCard really could be used to produce true hypertext. The advantage was that it already contained routines for making an index, map, and contents, and in particular for enabling the use of 'hot' words in the text that were actually linked to the word rather than superimposed over it[1].

The original text files from the prospectus were used, along with scans of the photos. The text was not marked-up in any easily accessible way, being in the form of PageMaker files. Utilities are available which can be used to convert hierarchical text structures into linkable structures, but were not available in this instance. This meant a laborious process of manual conversion, from PageMaker files to ASCCI files, and then importing into HyperCard text fields. This was time-consuming, and it proved necessary to keep a careful record of the files and their translated versions. A consistent labelling strategy proved vital.

The next task was to arrange it into appropriate chunks. This proved difficult. At first the text files were imported with their existing structure. The question was, how and where to organise the chunking? The choices were to chunk whilst the material was in ASCII format, or to import it into HyperCard and then chunk. Chunking whilst the material is in ASCII format had the advantage that material could be saved under separate and relevant file names, then imported into a single card field using a HyperCard import script. This only worked with whole text files, which meant that a large number of text files had to be created, one for each chunk of information. This meant even more care with file names to avoid confusion. Alternatively, the tactic

[1] This was before the addition of this feature to the standard version of HyperCard

chosen, existing relatively large text files could be imported into a temporary 'text files' stack, and then either chunked whilst in that stack, or cut and pasted in chunks from that stack into the actual prospectus stack. Again, this was a tedious process, complicated by the strain that chunking imposes in cognitive terms: the author has to make decisions about the information content whilst engaged in the mechanical and repetitive task of importing and copying across text.

HyperCard is capable of a variety of hypertext structures, but they must normally all be authored by hand. The underlying structure of HyperCard is linear, or more accurately, circular or looped. That is to say, that if no links are added by the author, it is only possible to progress from card to card by moving from one to the next in line, until one is returned to the start. The net result of these two phenomena is that it is easiest to produce a linear structure in HyperCard. By importing the files in sequence, the linear structure of the printed prospectus was replicated. Some efforts were made to add extra links, but the linear structure proved to inhibit this. The result at this stage was really nothing more than an electronic page turner with an active contents list and a few extra links as shortcuts to certain areas of the information.

This provoked a decision to author from scratch, using the pattern note analysis as a basis for the new structure. It was clear though that firstly a number of elements had not been considered by the participants, and that to proceed solely on the basis of the pattern notes would lead to a rather quirky form of prospectus. This was not a major problem in it had always been considered necessary to include elements from the printed prospectus. Secondly, it was clear that the elements included were not linked as one might expect: certain logically apparent connections were not made by the users, and some elements were totally unconnected. So the material in the structured diagram was modified to include obvious logical connections, for instance 'history of city' and 'character of city'. The information was also supplemented by developing a hierarchical structure for the information. A structured list of the elements was produced, and this was merged with topics extracted from the printed prospectus (P03, Appendix 2b). From this a hierarchical structure was developed initially on a bottom-up basis, with individual elements being sorted into categories until a top level of a few large categories was reached. This was converted into diagrammatic form (D01, Appendix2b). Links were then added across the hierarchy as indicated by the cumulative pattern note to give a cross-referenced hierarchy (D04, Appendix 2c).

5.3.2.5 Design specification

A formal design specification was not produced. For the second version, the structure diagram provided the main specification. This only provided details of links between nodes and node clusters. It would have been useful in some respects to have had a fairly detailed specification that indicated all contents, including navigation and access methods etc., but the structure diagram accompanied by some rough storyboards proved adequate for a work of this scale.

5.3.2.6 Information content: creation and input

For the main authored-from-scratch version, the material already imported into HyperCard was retained, and a linking structure imposed using the structure diagram for guidance. This saved re-importing the material. The shell was modified to give a short linear opening sequence to a main menu. This functioned as the first level in a flat hierarchy. Two kinds of card could then be created, using a scripted utility, part of the original shell. This gave head of section cards and content cards with a sub-heading. The sections corresponded with the second level in the hierarchy. Clicking the mouse on the contents item could allow a jump either to the head of a section or to subsections within it. Some of the text was edited and re-chunked to fit modified headings and sub-headings. The course description sections took a standard structure (See D01, Appendix 2c for details).

A further point is that it is clear that final decisions about chunking are to a large extent contingent firstly on the software package used, and secondly on the interface chosen. So in this case, HyperCard's frame or card-based approach meant a limit to the amount of text which could be displayed on screen, unless one uses a scrolling field which enables cards to hold more than their physical representation suggests. If one uses a scrolling field, one is limited to the rather graphically dull standard HyperCard scroll bars. Then interface decisions about for instance font sizes have a direct impact on the amount of text on a card. However, it is sometimes suggested that the user interface is a design task for later in the design process, for instance in Isakowitz et al's (1995) methodology. This makes some sense, and it is a common approach, but it does not always make practical sense. One at least has to make decisions about the area of the screen available for text, graphics etc. before deciding how to chunk. Then one has to consider the use of ancillary windows - pop-ups etc. - for extra material that does not fit, which requires at least some preliminary decisions about guidelines for allocating material to such items. A significant part of the information design concerns the details of representation of the various parts of the information, for instance the establishment of a hierarchy of headings and the techniques used for emphasis within the text. As similar typographic cues may be required for these and for link representation, it is helpful to make these decisions at the same time, and early, preferably before text is edited. There is also in any case a strong contention that user interface issues ought to be considered earlier on than is conventional in software engineering and information systems design, that sequencing activities is an over-rigid approach that does not fit readily with design practice (eg. Hix and Hartson 1993).

An alternative that is sometimes suggested is that if a chunk does not fit you should rewrite the chunk and make it fit. How practical is this? In some cases it makes sense, and this tactic was adopted for some items, but frequently it is not possible. For one thing, the original printed prospectus is quite economically written. It is difficult in some areas to see how it could be reduced. Expansion is sometimes possible, but not without materially extending the topic. There is also a question of the sponsoring institution's attitudes to changes in the material. Permission for this may not be forthcoming, or be slow.

A significant issue here is the point that it is essential to document one's materials in some detail. One of the key problem areas is keeping track of what is a) to be included b) available and c) actually in the author's possession. This is not a matter of a simple listing - it is necessary to have listings of what the raw material is, what elements of this are for inclusion in the stack, what the material form of these elements is at any given time (eg photographic print, scanned file with format, ASCII text file etc.), decisions made about what should be included, and in the case of materials which need to be converted from one form to another, some kind of key as to what original material has been converted into which new form. One of Shneiderman's (1989) 'Golden Rules of Hypertext' concerns such 'careful housekeeping' questions. This problem grows when there are a wide range of different media involved, when issues such as copyright and compressions also have to be considered.

The question then becomes, what is the best way to keep this record of materials? A set of listings in HyperCard has certain attractions - it would then be possible to relate the lists of files to the titles of fields etc. in order to use HyperCard's search facilities to check the inclusion of materials (but how would this work with graphics created/held within HyperCard?). However, this requires a certain amount of extra work to be done to best effect. Is it worth it?

HyperCard's linearity was useful in keeping a track on the material imported. The existing prospectus provided the sequence for importing, and then cards were created to reflect this sequence.

5.3.2.7 Linking and access

As indicated above, HyperCard has an underlying linear structure, so all nodes are automatically linked in the order in which they are produced. This may or may not be relevant to the designers' intended structure, and as discussed it can lead to an acceptance of an inappropriate linear structure. Here the existing HyperCard shell made some of the linking decisions automatic. A script in HyperCard's accompanying scripting language, HyperTalk looked up the titles of cards and entered them into an active contents listing, giving section and card titles which could be clicked on to jump to those items. It was then a relatively simple matter to use background buttons showing on most cards to allow a linear route through most sections. There were no extra access mechanisms at this stage.

5.3.2.8 User interface design

A number of conventional principles and guidelines for user interface design for hypertext exist, and an extensive body of knowledge. This material was examined in some depth for the taxonomy of design guidance. However, this research took place concurrently with the development of the electronic prospectus, so the final results were not available. Sets of interface design principles and guidelines for hypertext were already available however, and two of these, by Shneiderman (1989) and Hardman and Sharratt (1990) were used.

These principles were loosely followed. The user interface design was kept very simple in this version, partly out of a desire to follow the convention of minimising cognitive load on the user (Shneiderman 1992, McKerlie and Preece 1993), more particularly to reduce the design task to the minimum necessary given that there was no additional support. It was felt that the information structure was most important at this stage. A standard format was adopted for all cards containing primary information. The exceptions were title, help, glossary and map cards.

Problems arise in connecting the user interface design and the information design. How information is presented on-screen is closely connected with the structuring of that information. On screen information needs to be structured in such a way as to facilitate reading, searching and browsing that information. How this is done partly depends on the cognitive and other characteristics that the user brings to the screen, and partly on the inherent structures of the information. There is no point in designing an interface in such a way that it supports the user's qualities yet fails to present the information content in such a way that its internal structures are apparent. Such an approach is self-defeating in that structures on screen cut across those in the information content to create a 'cognitive dissonance'.

5.3.2.9 Testing and evaluation

Testing and evaluation was minimal, evaluation being limited to one or two colleagues and a few first-year students. This runs counter to the arguments of many in the HCI area, who suggest that user testing is paramount. It can only be said that, as others have found, user testing is time-consuming and difficult to arrange. It was not considered to be a major part of this study to investigate evaluation, this having been researched by a number of workers over the years.

5.3.2.10 Conclusions

This experience suggested that the key design problems in authoring hypertext for this domain were firstly in finding quick but effective task analysis methods.

Secondly was converting the existing structure into a worthwhile hypertext document. The linear structure of the printed prospectus tended to be retained in the hypertext, giving very limited hypertext functionality, to the point where the result was an electronic page-turner with a few added links. It is difficult when faced with the original document to make any worthwhile imaginative conceptual leaps, and one tends to lapse back into the existing structure. For this reason alone it is worth forgetting the existing document and abstracting the information, to restructure it in a more purely hypertextual way, concentrating on the information and task requirements, and structuring appropriately. The alternative is to use pre-determined structuring devices such as the active contents index in the Electronic prospectus, and to fit the information to that structure. This distinction is referred to by Hardman (1995) as between content-based and navigational structures. This approach has its attractions, but fails to use the capacity of hypertext to reflect the logical and semantic, or as Hardman (1995) calls it, the 'natural' structure of the information.

Text handling was a tedious process , and it was easy to import material twice, or to import the wrong material. Correct identification, with lists of the original file names, their ASCII conversions, and their final destination was essential to keep track of what material was where. Having imported text chunking was a major headache.

Another problem was that of the relationship of design to authoring software. Ideally the designed document should be designed for what the users and stakeholders require. In practice, they only get what the software will allow given the ingenuity of the design team. That being the case, at what point should one start designing for the available software? Prudence dictates that if resources are limited it is particularly important for the designer to know at an early stage what constraints the software is likely to impose. With a limited budget there is little room for going out and getting better software. There is also the question of learning and training time. On the other hand, the rapidly changing market for software applications makes it tempting to resist early commitment to particular software packages, and from a design point of view, design is likely to be freer and more imaginative if no constraints are set at first.

Finally, in terms of preunderstanding, the active involvement in design of an electronic prospectus was invaluable in familiarising the author more closely with the research domain, and more specifically made the task of designing interview questions for the case study easier, made conducting the case study interview more effective, and gave a set of concrete problems to relate to the task of creating a design methodology.

5.4 Later work

Later in the study, it was necessary to test elements of the methodology. How this was done depended on the element concerned. In some cases, it was necessary to work in conjunction with the electronic prospectus. This work was ad hoc, being driven by need and curiosity. It has consequently been described where necessary in the section on development of the design methodology, Chapter 7.

5.5 Implications for the design methodology

As explained in the methodology section, the main value of this part of the study has been in its ability to provide a first-hand understanding of some of the design problems and decisions faced by the design teams studied in the case studies. It has also provided a further insight into the value or otherwise of the guidelines etc. considered in the taxonomy.

6: Case Studies in Prospectus Design

6.1 Purpose of the case study

In broad terms, the case study is concerned with gaining evidence of the design process as conducted by practitioners. In this part of the research, the aim was to propose some general recommendations on the basis of the individual situations studied.

6.2 Research method

Case study research is generally considered as qualitative research, although there is no reason why it should not contain quantitative elements. Stake considers case study research as being not so much a methodological choice, but a choice of object to be studied (Stake 1994), and if that interpretation is followed, it is clear that case study research can cut across the boundary between rationalistic and naturalistic, holistic science. Much depends on the approach to case study work that is taken. Stake concerns himself with case study research in which qualitative inquiry dominates. Yin (1994) concentrates on an approach to case study research which attempts to reproduce aspects of positive, empirical research. The approach to case study research which is adopted should be determined by the research aims and the context in which research must take place.

6.2.1 THE VALUE OF CASE STUDY RESEARCH

The case study research method was chosen as the most likely method of obtaining relevant data about hypertext design. The particular advantage of the case study in this context was originally seen as its ability to get the most from a small number of data points. This was undoubtedly the case when this research was begun, and still largely the case at the time of writing.

As well as this particular advantage, case studies have other advantages, and some disadvantages. Case study methodology is not always highly regarded. Although used extensively in such disciplines as educational research (Stake 1995), it has frequently been seen as being problematic in terms of both validity and reliability, being seen at worst as an example of a single-point survey method with no grounds for generalisation to other examples or areas. The case study method has come under criticism in the past for two main reasons (Yin 1994). Firstly, case studies are potentially prone to a lack of rigour. Benbasat et al. (1987) found that a selection of reported case studies in information systems showed a number of faults, in particular failure to properly explain research designs, failure to describe data collection procedures adequately, and a failure to explain the reasons for the choice of case study sites. Reliability is thus reduced. Such a lack of rigour is not however an inherent fault of the method, rather a consequence of carelessness or bias on the part of the researcher. Secondly, it is claimed that there is a limited basis for generalisation, because in most case studies there is only one or a few instances. However, this depends on what case studies are compared to. If we think of the case study as a sample from a population, then clearly generalisation is difficult. However, Stake (1995) argues on the basis of Runkel's (1990) analogy that suggests that research consists of casting nets, ie. making comparisons across cases, and examining specimens. Case study research in this context is a form of specimen examination which enables extensive knowledge of the single instance.

Stake characterises, tentatively, three kinds of case study: intrinsic case study, instrumental case study, and collective case study. In intrinsic case study, the researcher is interested in the case for its own sake, because of its own intrinsic qualities. In instrumental case study, the case is of interest because it can extend knowledge and understanding of an issue or theory. The case is important only in so far as it can do this. Such a case need not be typical. The boundary between intrinsic and instrumental case studies is not fixed, but shifts as the researcher's interests and focus shift. The third approach, collective case study, is one in which a number of case studies are linked together in order to develop generalisations about a wider group of cases or a population. Such generalisation may require a further methodological framework, for instance Stake cites the sociologists Martin Kohli and Fritz Schutze as using Strauss's grounded theory approach. In an application closer to the current domain this approach has also been used by Davenport (1994) in a study of the use of IT by information science professionals.

Yin argues a more powerful case than Stake in seeking to place case study research within the rationalist-empirical tradition. He argues that the case study more closely resembles an experiment. An experiment is made and a theoretical proposition ensues from it. The experiment is then repeated, and hopefully the same result is obtained, although under new conditions. From repetition the theory can be confirmed. As Yin puts it, 'case studies are generalisable to theoretical propositions and not to populations or universes' (Yin 1994 p10). Although there are obvious differences between case studies and experiments, the two are clearly allied in that the case study is not a sample, and the aim of the case study researcher 'is to expand and generalise theories (analytic generalisation)'. This point is echoed by Mitchell, who defines the case study as "... a detailed examination of an event (or series of events) which the analyst believes exhibits (or exhibit) the operation of some identified general theoretical principle." (1983, p192).

The third objection to case studies is that they are lengthy and time-consuming, and consequently expensive. This is not an inherent quality of case studies but is dependent on factors such as the data-gathering approach(es) chosen, which is/are independent of the overall method.

Finally, case study research tends to be largely qualitative, although this is not necessary. As such it is vulnerable to the some of the same criticisms which have been made of qualitative research: it is subjective, slow and tendentious, and answers are less forthcoming than questions. Ethical problems may be raised by the tendency for such research to focus on the personal (Stake 1995).

According to Yin one uses case studies when it is important to cover contextual conditions. Case studies are also useful as mentioned above, where few data points exist and survey/sampling methods are therefore not feasible. A further benefit is that case studies are well-suited to the capture of extensive detail about reality, and dealing with a large number of variables (Galliers 1985).

6.2.2 OTHER CASE STUDY CONSIDERATIONS

6.2.2.1 The role of the researcher

The case study researcher is in a critical position. It is inevitable that the researcher will condense and select from the material available. To what extent this is done is the researcher's decision. The criteria for this are varied, and include the eventual readership, the purpose of the research, place and type of publication, the sponsor's needs, and many more. Also significant is the researcher's preunderstanding (Gummerson 1988).

6.2.2.2 Uniqueness and typicality

Generalising from case studies is problematic. Although theoretical generalisation is in itself valid, there still remains the question of the uniqueness of the case study. In many cases generalisation may turn out to be inappropriate or constrained. This in itself is perfectly acceptable. As Stake points out, drawing on the work of Feagin, Orum & Sjoberg (1991, in Stake 1995), and Simmons (1980, in Stake 1995), generalisation need not be emphasised in all cases. In the current context, this presents certain difficulties. To make design prescriptions of any kind, generalisation is involved. This question is considered more closely below.

6.2.3 THE CASE STUDY METHOD IN THIS RESEARCH

The case study method was considered appropriate in the context of this research particularly because, as mentioned above, the number of data points were apparently few. Very few HE institutions in this country were using electronic prospectuses, and other methods of acquiring data were therefore unsuitable. A further reason for the choice of this method was that it is considered very appropriate for exploratory research and for hypothesis generation. This study was not primarily concerned with hypothesis generation, but to propose design advice is arguably to make hypotheses about the best way to do something. This research situation can therefore be seen as paralleling situations in which hypothesis generation is a priority, and the case study method is therefore appropriate. A further point in favour of a case study approach within this study is that case study methods are particularly good at providing insight into the contextual and organisational aspects of projects in such areas as information systems design (Saurer 1993).

In the light of criticisms of lack of rigour, a failure to carry out and report case studies thoroughly, it is particularly important that the research design and process are described properly. The following section adopts the structure used by Sauer (1993) It first concerns itself with the research design, and the process by which the research design was executed. It then goes on to describe the protocol which was used, and gives details of the resulting database.

6.2.3.1 The research design

The central focus of this research as a whole was to generate some appropriate generalisations about electronic prospectus development, which could be incorporated in a design methodology. The case study part of the research was intended to enhance this process by providing data and some consequent generalisations on the basis of actual experience outside of that of the author. So what kind of data was the case study required to produce?

In this case the emphasis was initially on descriptive and exploratory work. The main concern in the first instance was to elucidate the methods that have worked, or failed, for existing developers, to find out what they did, and which of these things they would do again. After this it was hoped that certain generalisations could be made.

Yin proposes that the design of case study research subdivides into 5 components:

(1) the study's questions

(2) its propositions

(3) its units of analysis

(4) the logic linking data to the propositions

(5) the criteria for interpreting the data

This approach provides a framework which can be adapted to the electronic prospectus design context:

6.2.3.2 The study's questions

In this context, the type of research question posed is critical. In the case of this study, the object is ultimately to produce a design methodology. This is not in itself a question, but underlying it there is a question: what guidance is it possible to make about the design of electronic prospectuses? The other parts of the study are also attempting to answer this question. The author's own experience in system development is intended to produce prescriptive data. Preunderstanding (see Chapter 5) is a significant element here, and the experience of the author's own electronic prospectus was useful in evolving the questions asked of the case study, particularly at the lowest level (Level 1). The analysis of design guidance was useful in providing additional and alternative perspectives.

This high-level question corresponds with the kind of question Yin describes as a 'level 5 question', a question about policy recommendations, the inclusion of elements in a design methodology being effectively a statement about design policy. The question can be further decomposed using Yin's classification of levels as follows:

Level 5 questions (questions about policy recommendations):

What conclusions and design recommendations can be offered to future electronic prospectus developers? What design support should be provided, and in what form?

Level 4 (questions asked of an entire study):

What is the best, that is the most efficient and effective, way to develop a hypertext electronic prospectus or similar document. And by implication, what should not be done?

Level 3 (questions asked of findings across multiple cases):

Not applicable in this case.

Level 2 (individual case study level questions):

This level is most problematic, in that a number of alternative questions could be asked here. After consideration the following questions were adopted:

1 What design guidance and methods were applied by the developers of the system (if any)?

2. Which of these were useful, and to what degree?

3. What was the design process?

4. How were design guidance and methods applied in practice?

5 What design problems were experienced, and how were they resolved?

Level 1 (questions asked of specific interviewees):

This level refers to the specific questions relevant to particular interviewees. These questions are found in the case study protocol (Appendix 3).

6.2.3.3 Miscellaneous points

Other aspects that Yin considers of importance are:

- **The extent of control that the investigator has over actual events**

In the sites chosen, the projects have already been completed, so it was not possible to exert any influence over events. This ruled out options involving researcher participation such as the action research approach.

- **The degree of focus on contemporary as opposed to historical events**

These case studies could perhaps better be seen as case histories in Sauer's use of the term. They tend to concern past rather than ongoing events and processes.

6.3 The case study procedure

6.3.1 CASE STUDY PROTOCOL

The study began with the construction of a case study protocol based on the methodological approach described above. This is recommended by Yin as a way of improving reliability. It was initially based loosely on the kind of protocol suggested by Yin, but was adapted to fit the particular circumstances of electronic prospectus design. The protocol resulting (Appendix 3a) was used as a basis for the Glasgow study. It included: an overview covering the type of study and its purpose, issues in case study research - the rationale for site selection, the research questions under examination, definitions of unit of analysis and some relevant terms; field procedures; the case study questions, in their initial formulation, defined using Yin's classification of levels, and including specific interview questions; the sources of evidence; an analysis plan and guide for the case study report.

6.3.2 DATA GATHERING TECHNIQUES

Essentially any data gathering technique may be appropriate in a case study situation. Given that these studies were retrospective, it was anticipated that the main sources would be the products, secondary evidence in the form of documentation of various types, and the testimonies of the participants. The only planned data gathering technique was the interview. No decision was taken on the subject of group or individual interviews, as it was felt that this could be unnecessarily constraining.

It was hoped that there were would be some documentation in existence, either in print or as computer files.

6.3.3 CHOICE OF SITES

This was one of the areas singled out by Benbasat et al. (1987) as being under-explained in a number of case study reports. It is significant because the choice of site can have far-reaching effects, either enabling insight or introducing biases. At the very least, the choice of site in multiple-case studies is bound up with the kind of case study one is conducting. Case study sites may be chosen on the basis of whether one is aiming for 'literal' or 'theoretical' replications. In this case, finding and gaining the co-operation of a team working on electronic prospectuses was the major factor affecting choice of site.

Several alternative sites were considered but the only site which proved to be available was Glasgow University, which had produced a CD-ROM prospectus for the Computer Science Department.

6.3.4 PARTICIPANTS

The participants were a team of final-year Computing Science students at Glasgow University, who had begun a prototype CD-ROM-based electronic prospectus as a student project. This had proved sufficiently interesting for them to be commissioned to produce a production version of the prospectus for distribution. The original team consisted of five individuals, reduced to four in the later stages.

6.3.5 DATA GATHERING

Most data gathering took place during the week of 15th August 1995. A copy of the electronic prospectus was obtained and reviewed before the interview was conducted.

6.3.5.1 The Interview

For the Glasgow study, the interview was conducted according to the protocol question schedule. This took almost a full hour, the interview being taped. Only one participant, Paul, was available for interview.

The interview was conducted in the Department's Macintosh lab where the project work was conducted. This gave the opportunity to examine the prototype versions of the prospectus, and some time was spent in informal discussion of these. Although the interview was conducted some time after the original work had been completed, Paul's recall of the project appeared to be good. The interview was conducted according to the interview schedule in the Protocol (Appendix 3a). This was a semi-structured approach although with a high proportion of essential questions. The schedule was followed fairly closely, although there were several diversions.

6.3.5.2 Documentation

The other main source was the design team's documentation. The documentary data gathered was what historians consider as secondary evidence, originally created with some other purpose. The principal advantage of using such material in this context is that there is little opportunity for the researcher to intervene in the topic of study (Williams et al. 1988). A further advantage in this instance is that the documentation data can help to corroborate the interview data. The main disadvantage is that there is no way of assessing the biases of the individuals who created and handed over the material: what they considered worth writing about, worth keeping, worth handing over. There may be large omissions which bias the material. The documentation was extensive, and in the form of word processor files, over 3Mb worth. Fortunately, some of this was duplicated. What was left consisted of an interesting variety of reports, e-mail correspondence and drafts of content and structures.

6.3.5.3 Indexing

All items in the case study were indexed from the beginning to allow effective referencing. This includes both the data collected and the author's collations and interpretations.

6.3.6 THE ANALYSIS

Analysis in case study research relies on building a chain of evidence from the data back up to the highest level research questions.

In this case, analysis began with separate approaches to the three data sources.

6.3.6.1 Product analysis

The initial analysis concerned the product, and involved notes and diagrams based on several sessions of use of the product,. This was done before the other data were collected in order to be as fully informed as possible for this, in particular to maximise understanding during the interview.

The product was the Visual Open Day, a hypermedia electronic prospectus using a variety of multimedia forms. It was constructed using Director, a commonly used multimedia development package. The version changed from 2 to 3 during the course of the project. Director is a time-line-based package which uses a metaphor drawn from the film industry. It resembles HyperCard in having a scripting language, which was used extensively to make the product in question. Like HyperCard it is not designed specifically for hypertext use, but has a number of possible uses. The product itself was largely hierarchical in structure, containing extensive hypertext linking, and was in colour. The content was centred on the Department of Computing Sciences, with sections on the University in general and the city of Glasgow. There were a number of video sequences, usually featuring students or staff members talking about various aspects of the University. There were limited search facilities.

6.3.6.2 Computer documentation

The documentation material was first indexed and tabulated (Table GT03, Appendix 3c). This involved the creation of a table and entering each file, entering a number, the title as in the file name, document size in bytes, details of the dates created and last modified, a brief summary of the contents based on a reading of the document, and the contents type of document, the author, where identifiable, the intended audience, the relationship to other documents. The index number was added to the file name to enable easy identification. The contents type utilised three categories: text files, ie. files containing text and other material for the hypertext system content; documentation files, and working document files. Documentation meant all 'formal' documentation - reports, instructions etc. - as opposed to working document files which were all the other bits and pieces. Assessment work was included in this. It should be noted that these categories have a tendency to merge together - draft formal reports being very close to notes. The most significant distinction is really between the material in the prospectus and the material about the prospectus.

The dates were obtained from the Macintosh file information, which gives a 'Date first created' and a 'Date last modified' The latter is clearly less informative than the former, as any small modification is considered, whilst the date of creation marks the definitive opening of the document. How useful these dates are is another question. There is always the question of whether documentation was raised at the time the tasks referred to were undertaken. The existence of a computer file strictly only marks the maximum period over which that file was being actively created and modified, nothing more. It cannot for instance indicate the preliminary work which such a file may refer to, or the notes that were involved in that work. The existence of a file concerning a particular task does not necessarily relate to the time that that task was undertaken. However, it seems reasonable to make some tentative assumptions. In the case of content files, there is likely to be a close correlation between their creation and the time at which work on related areas of the project took place.

In general, the order of file creation is likely to reflect that of the work concerned. In some cases it was clear that file dates were broadly accurate because it was possible to build up a picture corroborated by other file dates on files referring to the same events.

For the most part the files enable some reasonable assertions to be made. For instance, it is clear that there was a distinction between the 'prototype' stage, the student project, and the completed VOD CD-ROM. This is reflected in the kinds of documentation and in the dates created.

The next task was document comparison. There were a lot of documents containing similar information. Documents were compared to see which contained the most relevant and useful information, and to see if the changes themselves provided any information about the conduct of the project. There were a lot of duplicate documents. In some cases later modification had added interesting material, whilst in others, later editing had removed it. All the files were examined and included in the table, even if they appeared to be identical with other material.

Authorship was often unaccredited, except in the case of e-mail messages and some versions of project reports. However, some of the logs for the first phase of the project provide some indication of who was responsible for what.

6.3.6.3 The Interview

The interview was transcribed verbatim (the transcript is included as part of Table GT05, Appendix 3b). This did not initially appear to be strictly necessary given the level of analysis conducted on the interview material, but as the interview diverged in several places from the original schedule, it was found to be essential. It also enabled the use of quotation during analysis.

The responses were transferred to a table (GT05, Appendix 3b) based on the original protocol schedule with the Level 1 questions. This showed the information to be elicited, the questions to be asked, the form in which they were actually asked, and the response. Some notes on the material were entered at this stage. As far as the questions asked were concerned, most of these appeared to have been understood, and for the most part elicited some useful and interesting responses, although the interviewee's responses were occasionally a little garbled and consequently unclear. It was also apparent that the interviewer had occasionally diverted from the schedule on the understanding that questions had been answered in earlier contexts when this was not the case. One or two questions remained unanswered as a result.

6.3.6.4 Analysis and synthesis

The first analytic step was to examine the interview to produce summary answers to the original Level 1 questions. These were also entered into table GT05. References to Level 1 questions and to the documentation were included here.

The next phase was a close examination and analysis of the various data elements. This was a synthetic process, calling for cross-referencing between the various data sources. There were two principal processes, with two outputs, a chronology, and a set of answers to the Level 1 questions.

6.3.6.4.1. Chronology

Firstly the documentation was utilised to provide a chronology of all design events, activities and tasks. Supporting evidence was obtained from the interview. This chronology was useful in providing a structure for further understanding, partly because it provided an organising principle for a large number of disparate fragments, and also because an understanding of the design process followed implies an understanding of what happened when. The chronology was developed by allocating documents on the basis of the 'date created' information from Table GT03, producing a table (GT04, Appendix 3d) showing a variety of events, tasks and problems with dates obtained from the file data, plus the sources of information.

6.3.6.4.2. *Answering the Level 1 questions*

The second analytic task was to use the material in Table GT05, the Level 1 answers, together with further supporting evidence from the computer documentation (Table GT03) to produce a set of answers to the Level 2 questions. This was done by relating the Level 2 questions back to the Level 1 answers, facilitated by copying the answers across into a new table (GT06, Appendix 3e) and adding further supporting evidence from the data sources as appropriate.

6.4 Results

The analysis above gave a set of answers to the Level 2 questions, which it will be remembered are the questions to be asked of a specific case. These are the first significant answers.

6.4.1 INTERPRETATION AND ANSWERS TO LEVEL 2 QUESTIONS

Below is the interpretation of material in Table GT05 and the chronology, as answers to Level 2 questions. This interpretation is based on Table GT06. References are to the individual Level 1 questions and to the computer files, the latter being numerals prefixed with a 'G'.

1. What design guidance and methods were applied by the developers of the system (if any)?

The design team did not use a formal design methodology (GM1). Certain techniques adopted from software engineering and systems analysis were adapted for the project, including a formal problem definition (GM15), requirements analysis (G005), feasibility study (GM6), structure diagrams (G016), an analysis of users (G004), and contextual analysis (G003). They used examination of other products to provide guidance as to what to do and not to do (G002, G045). Not all of these were formally completed at the time they were required - many were completed retrospectively.

There is some evidence of the use of user interface design principles (G012, G005), although the interviewee suggested that 'common sense' was the main guide (DD8). The principles were as follows:

- Design for ease and pleasure of use
- Design for consistency
- Design for a non-modal system
- Design for structural visibility
- Design for flexibility
- Design for direct entry to information
- Design for extensibility
- Design for reversibility of actions

The source is unknown.

In general, the design approaches taken appear to have been influenced by the team's earlier training in computer science and software design.

2. Which of these were useful, and to what degree?

It seems that the analysis techniques had a complex role: 'it wasn't linear, it was, you had to work backward and forward through it,' (GM6) - so some of the analysis was written up retrospectively, which suggests that such techniques as requirements analysis and specification were limited in their usefulness. The interviewee did not voice any misgivings other than this. This is of course broadly in keeping with other evidence of software - and other - designers at work. No structured analysis was used. The team concluded that:

"Although the structure of the system we are developing will be complex and of central importance to the overall end-product; structured analysis is limited because there is no data flowing through the system. Each screen full of information is displayed when the link to it is activated. There is no data processing taking place at all; other than referencing different pages. This makes the use of Data Flow Diagrams and other forms of structured analysis unnecessary." (G016)

The user interface design principles were somewhat general, and of a kind that is open to much interpretation, and it is therefore difficult to say for certain that they were followed to any great extent. The interviewee's reference to 'common sense' (DD8) suggests that perhaps they were not taken too seriously. Such principles as 'design for a non-modal system', and 'design for consistency' appear to have been followed (G012, CCD01). It is clear that they are also aware of the problem of 'cognitive overload' despite its non-appearance in their list of design principles (G012). This suggests the integration of a wider set of principles and guidelines on an informal basis. This is perhaps the basis of 'common sense'.

3. What was the design process?

The project started from an idea by Richard Cooper, a member of the Computing Science Department's academic staff (supervisor) (GC1). The clients were Phil Gray, also a member of the Computing Science staff, Catherine Lyons, their information officer, and a group within the department responsible for marketing and student recruitment, the INTERFACE group (G091, GM6). An initial problem definition was worked out (G098). This shows a four-stage development was anticipated, starting with a review of the existing electronic prospectus, followed by an early design, a prototype and user testing, followed by a full implementation using leading-edge technology. This was followed by a requirements analysis defined on the basis of analysis of existing software (GC10, C02), contextual analysis (G003), and input from Catherine Lyons and Phil Gray. A somewhat limited user analysis was conducted, which was based on personal ideas as ex-applicants, and contacts with schools and education departments in Scotland (G004). No formal task analysis appears to have been conducted. An analysis of the information content was made by examining an earlier version of an electronic prospectus for Glasgow University made in HyperCard. This appeared unsatisfactory, and the information content and structure was rethought. They assessed the relative balance of the various elements in the existing HyperCard stack, and weighed this against their existing ideas about the material (GC12). Then they integrated this with the idea of a Virtual Open Day that Phil Gray came up with, that is, they adapted the information content of the existing stack to fit in with this structure. A requirements analysis was produced. D005 shows a requirements analysis in terms of 'who, what, where, when, how' basis. This is followed by a formal requirements statement listing tasks the system is to perform; a system specification in terms of features, including some general principles and guidelines; functional and non-functional requirements, hardware characteristics, error handling, quality issues etc. There is no evidence as to how the design team was chosen, and one assumes that as it was a student project they were already friends and acquaintances. Information content and resources were based on the information analysis earlier. Choice of hardware was based on hardware available in Scottish schools - predominantly Apple Macintoshes (DR3, G031). A decision was made to use proprietary authoring software. This was based partly on personal interest - they were 'sick of hacking out code', and becoming interested in information systems as opposed to software engineering. It was also simpler to use the ready-made features of Director than trying to write routines for everything from scratch (DD 12, DD14). Software choice was initially constrained by the choice of the Macintosh platform, and narrowed down from a list of four products, HyperCard, SuperCard, Authorware Professional and MacroMind Director, to MacroMind Director, largely on the grounds of superior video sound and colour handling (DR4, G029). CD-ROM was chosen as the distribution medium on the grounds of the likely size of a document containing digital video, costs of production and reliability (DR5, G029). Director version 3 was used for the prototype, and version 4 for the second phase, the main VOD.

A number of other items of software were used during the project (DR7). These included PhotoShop for editing and manipulation of still images (G028) Simple Player, MovieShop, Fusion Recorder (G030) and Premiere (C027) for video capture, review and editing, Sample Edit for sound editing, Microsoft Word, Claris Works, PhotoCD (G026). A constraint was that there was only one copy of Director available. This meant that it was important to have alternative software for reviewing and editing video, hence the number of different video capture and play software items used (DR7).

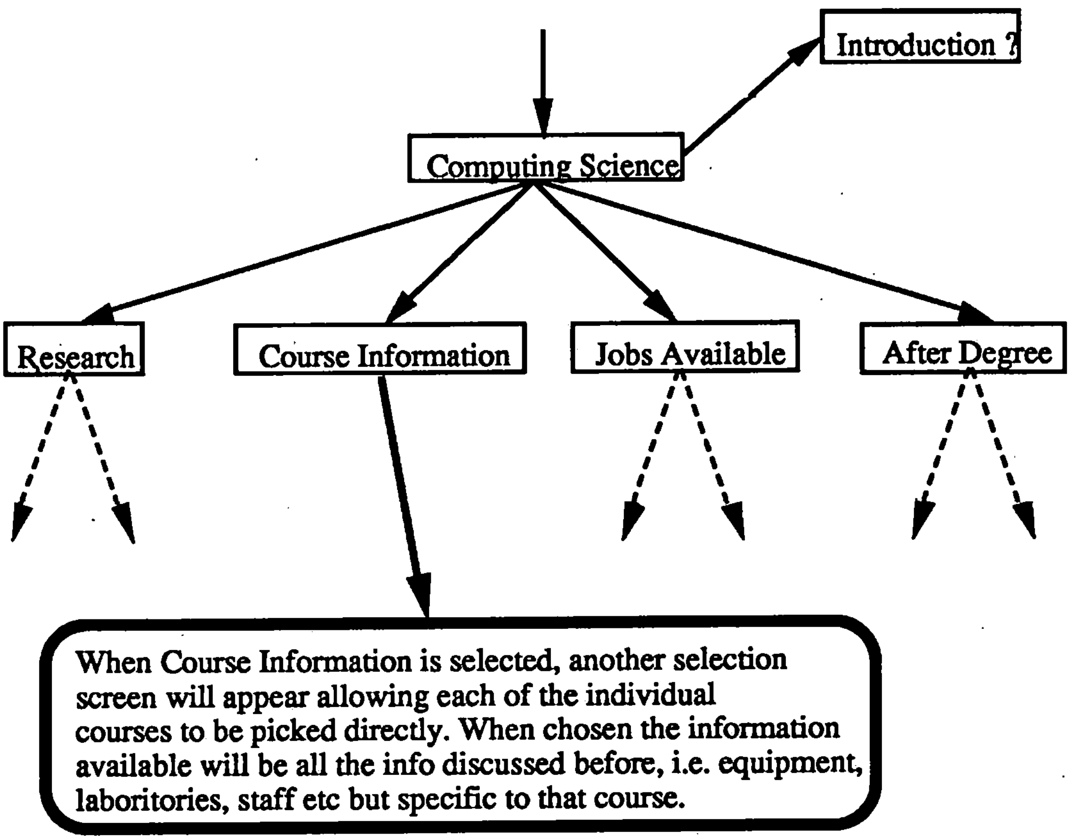
A limited version of prototyping was used (DR7, G032). There was an initial prototype, which the team considered as part of a feasibility study into the possibility of a full electronic prospectus. Much work was done on paper due to lack of availability of Director at first (DD3). This prototype was completed to the extent that all departmental information was included, but little general information. No extensive evaluation took place. The prototype was then largely rebuilt using the same basic concepts and many of the interface features (G086). In July 1994, work commenced on a full VOD prospectus, a more polished version with much more extensive material, including sections on the University and the city of Glasgow. At this stage, the focus was on the general feasibility of the project, especially in terms of technical possibilities and the kind of presentation of information that was possible (DD1)

Design resources were limited to design advice from Phil Gray (DR8), and the assistance of a photographer from the Media Department. There was no assistance with graphics or information design.

The information content was described as relatively unimportant in the prototype stage, as opposed to structure, appearance, navigation and control (DD1, DD3). However, the documentation reveals that it was considered as important in evolving an appropriate structure (G088). The two cannot easily be separated. In the second phase, information content was either authored by the design team or by Catherine Lyons. (DD1). General design ideas came from experience of other software (DD2), and from the team's exploration of the authoring software and manuals:

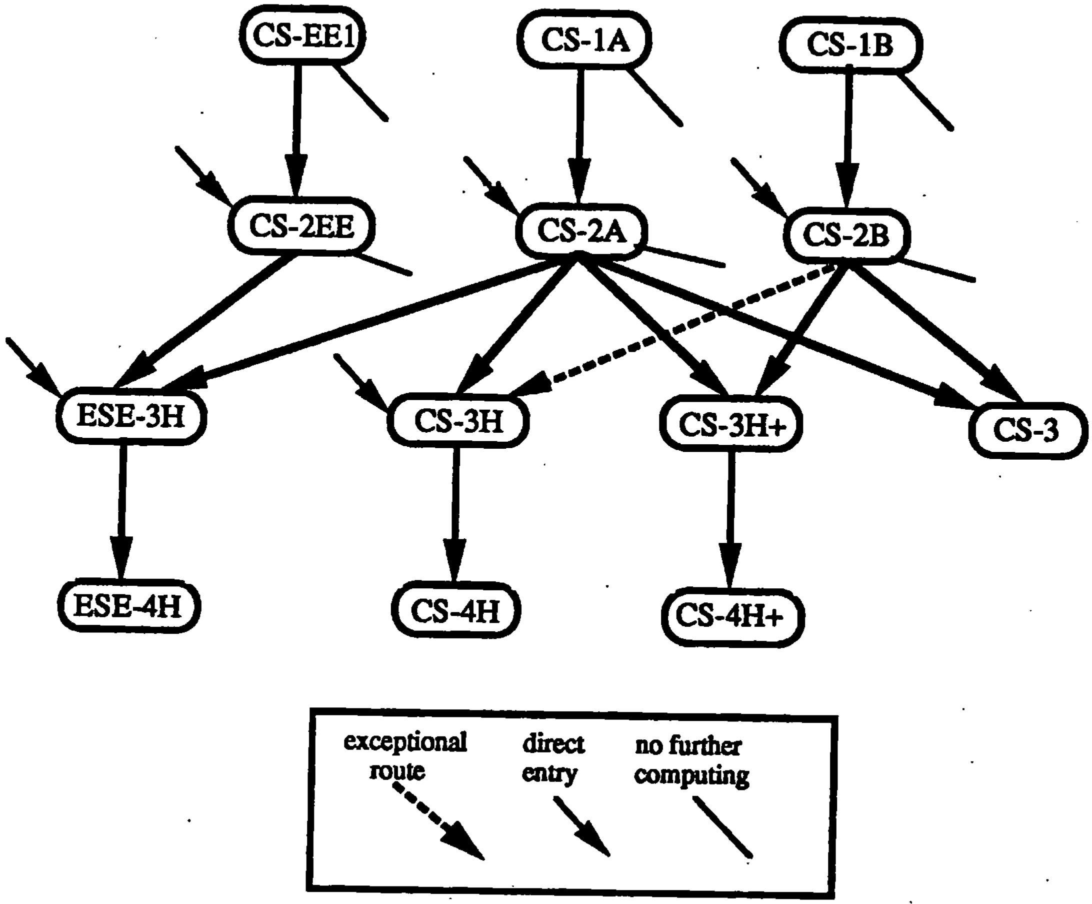
> "And someone's been reading the Director book: 'Oh we can do this as well'." (DD3).

The system was designed from scratch, and did not borrow from either the existing paper prospectus or the original HyperCard prospectus (GC10). The overall design approach was a top-down one (DD3), something which might be expected given the team's academic background. User interface design was considered at a fairly early stage (February - G007). A series of modules were designed using a hierarchical structure of main menu and sub-menus (G018, G092, DD5), and the information content and other material filled in. The exception is a rather more complex cross-linked section dealing with course options (GCD01, G043, G088). These are shown - and this seems to confirm the way they were designed - as modularised units and sub-units in G092 (Figure 6.1), and in more detail in G088 (Figure 6.2).



(Figure 6.1) Course information module structure (G092).

Undergraduate Courses



(Figure 6.2) Course structure diagram (G092).

This series of diagrams was the main design technique. The rationale behind this choice of structure was to simplify navigation, which the team had seen as a problem with the original HyperCard stack (C002). Apart from this, the indications are that design was a fairly flexible and loose process. There appears to have been no design documentation for the second phase, the construction of the main VOD (DD18).

The second phase proceeded by section. The document was structured into several sections made from separate Director 'movies' linked together. Skeleton structures were made for each module, then filled in (C2). The Department section was authored first, because it was most difficult (C1, G048, G052), followed by the City and University sections, which were worked on concurrently with the scripting for the Department section (C1, G106-G108, G110-G 112, G135).

The VOD prospectus was evaluated whilst the second phase version was in a part-finished state (DD15). Evaluation took place via a questionnaire and protocols conducted using prospective students at the Departmental Open Day, with the aid of psychology students (DD15, DD16, G013-015 - results, questionnaire and protocol tasks). The main priority was ease of use (DD16). Tasks and questions focused on navigation issues, picture quality, movie quality , effectiveness and value as conveyors of information, use of sound in gaining users' attention, general graphical appearance of the system, general effectiveness of the system, and suggestions for improvements and general comments on the system. This revealed a number of problems, apparently mostly relating to movie handling, although the related document is incomplete (G015). There was also some informal evaluation via schoolchildren brought in by Catherine Lyons (GC9).

Documentation was extensive but somewhat unstructured (G001-153). Some elements were well-documented, for instance there was a fairly detailed log for the first phase of the project (G042). There was a report to the INTERFACE group at an early stage, a project report at the end of the prototype phase, and after that documentation is limited to working notes such as notes on colour codes (G072), formatting decisions (G070), problems and un-implemented ideas (G069), and bugs and missing items (G113). No master documents were maintained for files, links etc..

Debugging was partly based on the testing at Open Day, and allowing stakeholders and other members of staff etc. to play with the system. Also the separation of the team into different sections enabled a fresh view of each other's sections that showed up faults (C6a). Lists were made of some of the problems and un-implemented items, eg. G069, G104 and G113. Some evidence of link checking exists, for instance in G104, but there is no evidence of a systematic approach to this.

4. How were design guidance and methods applied in practice?

Storyboards and structure diagrams were evolved before the Director software was available (DD3). Several structural options were worked out in diagrammatic form (G091).

User interface design principles were incorporated into the requirements document (G005). It is not clear how they were utilised in practice, but the interviewee's reference to the use of 'common-sense' rather than any specific principles suggests they were not used rigorously (DD8).

The requirements (G005) document does not appear to have been used as effectively as it might have been. It is comprehensive and contains some quite specific requirements, which could have been used as the basis of evaluation, but the evaluation tasks and questionnaire do not directly correlate with the requirements. Requirements documents may also provide a useful checklist when evaluating authoring software.

5. What design problems were experienced, and how were they resolved?

A number of problems were encountered. To begin with, the software was not available, and it was difficult for the team to know what was possible:

> "So it's kind of difficult, because you've got this sort of like an information gap between knowing what you want and not being sure what Director will actually do." (DD3).

Navigation was an early concern, navigational problems being apparent in the old Glasgow electronic prospectus (G002). The hierarchical structure meant it was possible to lose track of location in relation to the rest of the system (DD10). The response was to add a locational device in the form of a path-name at the top of each screen, and to constrain the options as much as possible.

A number of small-scale technical problems were encountered. These include problems with page transitions, keeping the Director score and cast tidy, and setting up the design so it could easily be changed (G069). Also problematic was the layering of objects. This did not appear to be amenable to a structured approach (C1). The structuring of the document into different movies for each section caused problems with sound continuity.

Both the hardware environment, in terms of screen size, and Director imposed certain constraints in terms of the amount and size of text and graphics that could be shown on screen. Problems in chunking material were not directly referred to although this might be an expected consequence of such constraints.

It had originally been intended to include a full search engine, but this was also found to be technically too difficult to be accomplished in the available time, so the search facilities in Director were utilised. This made it necessary to anticipate in advance the kinds of things users might wish to look for (DD12).

The most difficult parts were the Courses and Guided Tour sections, the latter being a section which utilised a lot of video and had quite a complex structure which caused a lot of problems with the users' routes (C1, G104), and with scripting. There is a trade-off between the desire to keep the script as tight and compact as possible, and therefore faster to run and less bug-prone, and ease of implementation (G104):

> "...I am thinking mainly of the course map to general pages - this will involve disabling all the sprites that represent the guides; supplying some default context and removing the signpost icon. It seems much easier just to duplicate the info (and the layout?) elsewhere." (G104).

Video was identified as a significant problem area, as already noted. Many video problems related to how to give users control. This led to a number of arguments with regard to how the user should be able to control the video sequences, in particular making sure that videos reset to the beginning. The options were to use the standard Apple video controller or a tailored version. In the end a tailored version was chosen (C3). Another aspect was the problem of getting video compressions right - getting an acceptable combination of utilisation of disk space, smoothness of movement and detail of image, and getting this right for the lowest specification of machine - in this case the Macintosh LCII. This entailed running all video on an LCII to ensure success.

Sound was relatively limited in the VOD prospectus, one reason for this being the lack of a sampling utility which could deal with longer sequences (G034). Another was the failure to gain the co-operation of the Department's sound specialist (G034). It was otherwise fairly straightforward except where the VOD was separated into different 'movies', although its potential to annoy was noted and a 'music on-off' button included for the developers as much as the users (C3).

Other problems were organising the team and allocating work, and version control - keeping everyone working on the last or best version of the project (C6). Another point is the learning curve associated with the software packages concerned. Where multimedia elements are concerned, the designer may be called upon to use a number of fairly substantial and complex software packages, and this was evident in this project, where the team had to use Director and Premiere. The reluctance to engage with another complex piece of software meant that in the prototype stage video was simply cut and pasted in from the video capture utilities without any editing (DR7).

In summary, it seems that there was limited outside expertise directly involved (GC4). This created a context in which problems had to be resolved by the team or features could not be implemented, as in the case of sound. Within this context, the main problems appeared to be technical although it is possible that this may be a distorted perspective, a consequence of the team's scientific/technical orientation as computing science students. These problems were frequently caused by the limitations imposed by the Director software. Choice of software is clearly critical. Although at the time the VOD was made, Director was probably as good a choice as any for this kind of project, it obviously posed many awkward problems. Given the present state of authoring software, it is probably a question of choosing which problems the author wishes to deal with: there are no perfect solutions. It appears that design is most severely constrained in the first instance by not knowing what is genuinely achievable with the authoring software available, and then by the various technical obstacles imposed by the hardware and software.

6.5 Interpretation of results

6.5.1 GENERAL:

They revealed a design process that was as much dependent on co-operation, experience and responsiveness to external constraints as to any particular development methodologies or guidelines. In particular, no specific development methodology was used, although there was a general adherence to software engineering and the only use of guidelines or other such advice was a short list of HCI principles culled from software engineering literature. Limited user testing took place, but much of this seemed to be for assignment purposes, rather than from any attempt to introduce iterative prototyping methods.

6.5.2 SOME REFLECTIONS AND CRITICISMS

As the discussion earlier indicated, case study research needs to be done carefully if it is to maintain an acceptable level of validity and reliability.

6.5.2.1 Validity and reliability

Validity concerns the degree to which the measure or research design operationalises the phenomenon or concept it purports to measure. There are two main aspects to this: internal validity and external validity. Internal validity concerns the internal logic of the measure or research design: the extent to which it operationalises what it claims to, and to which it excludes alternative explanations or hypotheses. External validity concerns the extent to which useful and meaningful generalisations may be made to other cases, populations and situations. (Williams et al. 1988).

6.5.2.1.1 Internal validity

Williams et al identify a number of potential threats to internal validity. Relevant ones in this research include such aspects as history, maturation, demand characteristics, experimenter bias and response bias.

History

This concerns the way in which features of a new medium may alter as a result of technical changes during a study. This may have a role to play: the project studied took place in 1995, and the available technology has improved. However, the basic approaches and interface designs of software such as Director remain largely the same, and it is reasonable to assume that most of the conclusions made regarding the case study are still applicable. The principal changes concern such aspects as video handling capacity, which is much improved, and the increasing orientation of authors towards the World Wide Web - where in fact such aspects as video are in a state resembling the early development of digital video.

Maturation

This refers to the way in which users gain experience with new technology over time. It is likely to apply in this context: people coming to hypertext development will be increasingly sophisticated with regard to understanding of and experience with hypertext and hypermedia. Some of the problems experienced in the case study may already appear to be trivial to experienced authors, and seem less to those inexperienced with hypertext but with a greater familiarity with computer hardware and software generally.

Demand characteristics,

This concerns the possibility that respondents may want to present certain characteristics to the interviewer, for instance to be more innovative with a new technology than they actually are. This is a potential problem in this case: the interviewee was an undergraduate and may well have been keen to impress with a level of knowledge and expertise beyond his real knowledge and abilities. There was little evidence of this during the interview however. If anything there was a tendency to understate what was done. The use of the computer documentation, and the existence of the VOD itself, acts as a way of checking for any bias here.

Experimenter bias

Experimenter bias occurs when the researcher brings assumptions to the research situation that cause distortions.

Response bias.

This can occur when the respondent perceives that a particular kind of response is appropriate to a question. There is always a question as to what extent can the material gathered be taken at face value. So how can the researcher be sure the interviewee is telling the truth? A classic exploration of this issue is that of Dean and Whyte (1958). They suggest four factors which can be expected to influence the informant's reporting of a particular situation under particular interview circumstances. These are ulterior motives, bars to spontaneity, desires to please, and idiosyncratic factors. Looking at each of these factors in turn, ulterior motives are unlikely to be significant. This researcher was not in a position to aid the interviewee in any significant way. It was made clear to the participant that the study was not for any assessment or evaluation by the University or any organisation connected with it. Bars to spontaneity should similarly have been minimal, the most likely problem being that the room was not private, being a computer lab. However, the interviewee appeared quite relaxed in this situation, and was aware that the information obtained was confidential. Desire to please may have had some impact, but it seems unlikely that this would skew the data obtained in any particular way other than perhaps an urge to say something rather than nothing. Idiosyncratic factors are difficult to detect, but the relatively large number of questions covering a fairly narrow domain should offer some protection against this.

Other forms of distortion may occur (Dean and Whyte 1958). The interview took place some seven months after final completion of the project. Recollection may simply be poor or the informant not have the required information.

The existence of the documentation in general acts as a protection against such problems as respondent bias. However, this still requires some care. Response bias of a kind applies to more than just the interview data in this case, as some of the work involved assessment for a degree. This gives a context for presenting a particular kind of public face, for instance convincing tutors that things are on schedule, that in-depth analyses utilising other course material have been conducted, etc. Hence there is a question-mark against the integrity of a few items in the June material, much of which concerns earlier tasks and has clearly been written for assessment purposes. Material for the INTERFACE report is also intended to create a favourable impression with the client, and may similarly be biased. However, in most cases other documentation exists which can provide corroboration. The project later became a paid project, and the conditions therefore change. Noticeably, the level of documentation tails off: most of the later material is directly concerned with the development of the project, as text files for instance, or as lists of bugs and things to implement.

6.5.2.1.2 *External validity*

As noted above, external validity concerns the extent to which useful and meaningful generalisations may be made to other cases, populations and situations. This has long been a concern with case study research, as was noted in the earlier discussion of the advantages and disadvantages of case study research. The concern is with the uniqueness of the case: to what extent can valid generalisations be made? This is especially true where there is only one case. The group studied may not be representative in that their background was in computer science, but the way in which they bring their own background and preconceptions to the project situation is likely to be typical. Adjustment can be made for this. Their relative naiveté in some respects makes them to some extent atypical, but again it is likely that many new designers of hypertext will show shared characteristics.

Two key areas are the choice of participants, and the analysis. Choice of participants affects validity. In this case, there are several aspects to the choice of participants which affect the extent to which it is possible to make generalisations from this work.

Firstly, their student status is bound to have an impact. They were engaged in a student project to begin with, with a different set of working conditions and goals from those held by salaried staff or contract workers. On the other hand, the aims are broadly similar: to produce an artefact, and the working conditions and materials are also likely to be similar.

Secondly, they were studying computer science. This is likely to have imposed a particular mind-set on the students. This is exemplified at various points, for instance, their reason for choosing the project was that it 'made a change from hacking out code', suggesting that the project was a fun activity as opposed to the logical rigours of programming. This suggests that they would have a positive attitude to the project which might not be shared by a salaried worker or workers. A further point is that their approach to the work showed a concern with hardware and technological issues, reflected for instance in a desire to show what the technology could do, rather than to satisfy the needs of stakeholders.

The point should be reiterated that generalisation here is not to a population. Other kinds of generalisations are possible. It is essential to be clear about what generalisations are being attempted. In this case it is suggested that the extent of generalisation is relatively limited, and that within this context the generalisations which are being made are reasonable.

A further point is that generalisation is not the only issue. Case studies can do more than offer generalisations. In the pursuit of this research it is also necessary to uncover possibilities, alternatives. The value of the case study in this context is not only in enabling generalisation.

In this case the rationale is that the tasks involved are likely to be common to all attempts to author such systems. All hypertext authors of electronic prospectuses are likely to have to engage in such tasks.

6.5.2.1.3 *Reliability*

Reliability means ensuring that the measures used would always give the same conclusions regardless of observer and situation. Whenever human observers make measurements of any kind, there is a risk that the resulting data be reduced in quality by random and bias errors. This case study is particularly prone to these problems, as only the one observer has been involved. A typical approach to controlling reliability is to ensure that measurements are taken by more than one person. This method has not been used in this case, partly for practical reasons and partly because re-interviewing was not appropriate. Each interview is a unique situation, and to re-run the interview would be to do something different. No genuine overall improvement in reliability would result, although some old data may be confirmed and new data might result as a side effect. The main source of reliability in this case is the use of several different sources of data. Material in the interview can for instance be confirmed by the existence of features in the CD-ROM, or by evidence in the computer files.

As far as reliability is concerned, the use of a protocol means that it should be possible for another researcher to get a similar set of data given the same case. What is more problematic is the question of the analysis. This could hypothetically have a considerable bearing on the reliability of the research, because much is left to interpretation by the researcher, despite efforts to use the protocol to prevent this.

6.5.3 Were the right questions asked?

One area that the study failed to consider is that of what methods and techniques the design team would have liked . This suggests a further Level 2 question to be asked:

6. What design guidance and support would developers like in future?

This could be decomposed to:

What improvements could be made to the existing procedures used?

What new guidance, support and techniques are required?

6.5.4 Level 4: overall findings

The case study cannot claim to give a definitive answer to the question 'What is the best, that is the most efficient and effective, way to develop a hypertext electronic prospectus or similar document?' And by implication, 'what should not be done?'. The points above explain why. However, some points may be validly put.

Software choice must be properly considered. Many of the problems experienced in this case were a consequence of the authoring software used. In many cases where problems arose in this project, it seems that they were resolved by constraining ambitions to fit in with the hardware and software.

Because of the steep learning curve associated with the use of different software packages, a range of experience (within the design team) of different software packages is likely to be very useful.

Thirdly, video causes many problems. Despite its many attractions, video poses technical problems in terms of quality and compression. It also poses manipulation problems for the user, which in turn create significant difficulties for the designer.

Careful version control has a role to play in the effective management of the design process.

6.5.5 Level 5: recommendations

Some general recommendations can be made on the basis of these points:

1. Unless extensive in-house programming support is available and a choice to produce a custom-built document is taken, software should be carefully investigated, and a fully informed choice made. This should involve a systematic procedure for evaluation, preferably based on a clear requirements specification. There are no easy solutions here though. It is not enough to say that the software should not impede the design solutions chosen. For the time being they will. The point is to minimise such effects.

2. Video should only be used if it provides genuine added value as its current technical imperfections are otherwise likely to cause problems out of proportion to the likely return.

3. Design teams should either include members with a range of software skills or should have a clear programme for the acquisition of such skills, including a strategy for an appropriate division of labour.

4. Close attention should be paid to version control, with a systematic record of versions being maintained.

7: Creation of a Design Methodology

"The search for a perfect methodology, it is argued, is somewhat illusory". (Avison and Fitzgerald 1988).

7.1 Introduction

The previous chapters examine how a range of data was gathered and collated to provide information on which to base a design methodology for hypertext. This chapter describes how these data were used as the basis of a design methodology, and discusses some of the issues and difficulties encountered in the process. It then outlines the methodology itself.

7.2 Rationale

It may be remembered from the literature review that the original motivation for another design methodology was the perception on the basis of the existing body of design methodologies and other design guidance that there was a gap between the more formal design methodologies such as RMM (Isakowitz et al. 1995) and the simpler design advice of Shneiderman (1989) and others. Given this perceived gap, what kind of design methodology might fit this gap, and how might such a design methodology be derived? The gathering of a range of data which may help to construct such a methodology has been described. This section describes the rationale behind the construction efforts

It is unlikely that any single correct way exists to design a hypertext document. Much depends on context and constraints: individual expertise, resources available, the demands of clients and markets. Given that this is the case, it is still arguable that there is a problem area which is open to better design approaches. It is first necessary to define this problem area, then to look at the notion of design methodology: what it is, and what a hypertext design methodology might consist of.

7.2.1 THE PROBLEM AREA

The emphasis here is on the techniques and procedural steps involved in the planning, analysis and design of hypertext documents. Although such issues as design team management and cost estimation, and the design of application programmes are clearly relevant and important, they are in themselves major subjects and cannot be considered in the current context.

7.2.2 WHAT DO WE MEAN BY DESIGN METHODOLOGY?

It is useful to remember what is meant by the term 'design methodology'. The term is most commonly found in information systems, a field which as discussed earlier, provides a source of relevant information and theory. Jayaratna (1994) points out that the term is essentially misleading, the term 'methodology' really being only applicable to the study of methods for scientific research, its more familiar meaning. What is really meant by the term 'methodology' here is the individual methods employed in design. However, the term is now widely accepted and used. Avison and Fitzgerald (1988) define it in terms of:

> 'a collection of procedures, techniques, tools and documentation aids which will help the systems developers in their efforts to implement a new information system . It will consist of phases, themselves consisting of sub-phases, which will guide the systems developers in their choice of the techniques that might be appropriate at each stage of the project and also help them plan, manage, control and evaluate information systems projects.'

This provides a reasonable general description of the main ingredients that might make up a design methodology for hypertext, but does not give much specific detail as to what such a methodology might contain. The initial conception, based on the author's previous experience in hypertext development, and on the literature review, was of a set of procedures which would guide the various design tasks: initial analysis, design of the information and the hypertext structure, the creation of nodes and their linking, the design of a user interface, and evaluation. Inevitably greater understanding of the subject area led to modification and extension of this conception.

When this research was originally conceived, it was unclear as to what the process of making a design methodology from the various elements, the taxonomy, the electronic prospectus and the case study might be. A process of synthesis was imagined, a merging of the various elements into a coherent methodology. From this somewhat imprecise conception it is a short step to view the process of design methodology development as a design problem in itself, broadly consisting of the classic stages of analysis, synthesis and evaluation. The previous chapters concern the analysis phases. The making of the methodology involves synthesis. Synthesis, in the design sense as opposed to the Cartesian sense, means a transformation of the data gathered in analysis into a more structured form which can be used in hypertext design. This is what Jones (1980) would describe as a process of convergence after the divergence of analysis, convergence that is of the multitude of design guidance and other data and recommendations to the single design methodology that is the desired end of this research.

The taxonomy of design guidance enabled a clearer understanding of the hypertext design process and the tasks therein, and a clearer notion of the relevant elements. For instance, on the basis of the design methodologies included, one would expect the typical ingredients of an overall set of procedures, coupled with a set of techniques and possible software tools. These in the hypertext context would be more closely tailored to the design of hypertext documents. Some techniques for integrating task analysis into hypertext design and translating the results into hypertext structures would also be expected. The existing hypertext methodologies offered a framework of procedures which it seemed could usefully be supplemented by some relevant and tailorable interface design guidelines for hypertext, including some guidance on how to manage the use of such material effectively. Documentation aids for some of the tasks identified in the literature as awkward such as link management, could be included.

Of course, the fact that existing design guidance emphasised these matters is not enough to conclude that they were the only things that mattered for a hypertext design methodology. The case study suggested that in many ways the concerns of design guidance writers had been correct; however it also suggested that the principal problems faced by hypertext authors were related to the technical difficulties manifested in translating designs into reality, or, in systems design terms, moving from logical design to physical design. This in part reflects the software available for hypertext authoring, and no methodology is in a position to improve this aspect. However, good design support can help by freeing the designer to get the most from the available software. The author's own experiences with the hypertext electronic prospectus suggested that the constraints imposed by the software were important, but that by designing with a good knowledge of these constraints, the problems could be minimised. Key problems lay first in extracting a valid information structure, and then with ensuring link integrity.

7.2.3 THE PHILOSOPHY BEHIND THIS METHODOLOGY

Behind every design methodology lies a wealth of assumptions about the world, the organisational context, and about social relations. It helps somewhat in judging the value of design methodologies to have some understanding of the underlying philosophy or stance of the authors of the design methodology (Jayaratna 1994, Avison and Fitzgerald 1988). So the following is an attempt to represent some of the author's ideas, assumptions and, probably, prejudices.

In this case, the author held the initial assumption that the authoring of hypertext was, at least potentially, a human activity like that of writing which should not remain as the narrow fiefdom of a community of experts in the computer sciences. Whilst the authoring of text documents is still not an activity open to all, it is a widespread activity that presumes no special disciplinary expertise. We have a long tradition of writing, and it is a highly regarded skill that is found throughout the population. There seems to be no reason why, given contemporary access to hardware and amenable interfaces, the writing of hypertext should not have equally wide currency. So the concept of a design methodology is seen as having a liberating effect in enabling a wider range of people to work effectively with hypertext.

It follows from this that a design methodology should therefore be accessible. It should not expect knowledge of computer science or similar disciplines or esoteric techniques. Several of the current design methodologies for hypertext like RMM (Isakowitz et al. 1995) take a computer science or software engineering paradigm, which assumes that the author will naturally use techniques from database design or systems design, such as entity-relationship diagramming. There is no reason why these techniques should not be employed, but it is the author's contention that such techniques are likely to meet with resistance from practitioners outside the domains indicated.

A further presupposition is that iteration and prototyping is an essential part of document design. However, iteration is adopted on the basis that, as Andriole (1989) suggests, prototyping is an admission of a failure in terms of requirements gathering, and that as such successful prototyping must be cost-effective. Therefore it is not considered that such iteration should necessarily involve extensive user trials, despite the undoubted value of such work. It is not possible, regardless of how desirable it may be, for many potential hypertext authors to do extensive testing with users, especially for the smaller documents envisaged as the target of this design methodology. Similar arguments for testing printed documents with users could be made, but such practice is frequently regarded as a luxury. The situation is not of course identical. There is considerable expertise in the publishing world in deciding what it is that readers want and how this should be done. Hypertext as yet lacks this body of expertise in the hypertext design community. Hence guidelines are seen as having a potential role in the design methodology, both for work in progress and in evaluation.

7.2.3.1 Some design methodology elements

Design methodologies typically contain a number of elements:

7.2.3.1.1 Models of design

Avison and Fitzgerald (1988) suggest that all methodologies adhere to a model, a view of reality , an abstraction. This enables the description of complex real-world phenomena and their translation into physical systems. This is typically seen in terms of a conceptual model, a logical model and a physical model. In design terms, the move is from descriptive models which describe the domain of interest to prescriptive models which outline the system to be produced.

7.2.3.1.2 Procedural stages and steps

Each design process consists of a number of stages and/or steps. Design methodologies attempt to define and organise the various design stages and steps. At their simplest, these are the stages of analysis, synthesis and evaluation. A typical extended list of stages in an information systems life cycle is given by Olle et al. (1988):

1. Strategic study.
2. Information systems planning.
3. Business analysis.
4. System design.
5. Construction design.
6. Construction and workbench test.
7. Installation.
8. Test of installed system.
9. Operation.
10. Extension and maintenance.
11. Phase out, and
12. Post mortem.

Not all of these stages need be the concerns of design methodology. For instance Olle et al. (1988) identify stages 2, 3 and 4 above as the central concerns of information systems design methodologies. They may also be defined very differently. For instance, many methodologies identify a feasibility stage at the beginning of the design process, such as the original National Computing Centre design methodology (Lee 1979), which describes the following stages:

1. Feasibility study.
2. System investigation.
3. Systems analysis.
4. Systems design.
5. Implementation.
6. Review and maintenance.

Differences in the role and function of a hypertext design methodology as compared to that of an information systems design methodology mean it is not possible to translate material in the field directly to that of hypertext document design, but such terms as those outlined above are recognisable generic terms for hypertext design tasks, and it seems reasonable to base the beginning of a design methodology for hypertext on such stages.

7.2.3.1.3 Techniques

Design methodologies generally utilise a range of systematic techniques to support their various procedures. Many of these are ways of representing models of the design situation. These vary in formality. They include such approaches as data flow diagramming, data dictionaries, entity models, and entity-relationship models. They often involve visual representation of the design problem. Many of these techniques are oriented towards information systems for organisations, in which data flow is a critical element, and are therefore often not particularly useful for hypertext design. Despite this, a number of techniques have been adapted for hypertext design, for instance Isakovitz et al. (1995) employ entity-relationship modelling. Other techniques sometimes used include various data structure and flow diagrams.

Requirements analysis

Requirements analysis is a generic heading for a number of techniques involving stakeholder, user and task analysis. No specific techniques for requirements analysis were recommended, but a number of techniques suggested.

Task analysis and user analysis techniques

There are a number of problems with task analysis, and a number of solutions have been proposed (Dillon 1994, Johnson 1992, Diaper 1989, Jonassen 1989). When the task domain is relatively constrained, the available solutions are fewer. A number of techniques suitable for hypertext task analysis have been suggested (Jonassen 1989, Dillon 1994).

Modelling and representation methods

The hypertext document has a number of aspects which may be usefully modelled. The task, the user, the information content and the hypertext document itself are all open to modelling.

Graphic methods tend to be most popular. They can represent complex domains easily, and are more readily understood by stakeholders than techniques such as formal notations. A number of graphic models were examined for the methodology, principal among them being structure charts, flow charts and data diagrams of various kinds such as data flow diagrams. The reasons for this are firstly that written documentation, whilst useful, is limited in terms of its ability to represent the design situation effectively (Bellman and Suchmann 1990). It is difficult to represent all the information for a complex document without repetition/redundancy, and the task of organising a complex body of information without visual aids can be very difficult. Secondly, the ambiguities inherent in language make it hard to ensure that requirements are translated accurately into systems. Graphic methods also possess ease of communicability to non-specialists as opposed to formal notations. Visual aids for design can be considered as forms of graphic models. They are abstractions of reality which in many ways simplify the real situation.

These concerns translate readily to the hypertext domain. It is certainly the case that hypertext documents may be complex, and the limits of written forms for describing hypertexts are soon reached. A hypertext structure can be described verbally in terms of lists of what is linked to what, but it is very hard to visualise the structure of the document without some form of graphic representation. The domain is likely to be more constrained than that of many information systems, but the risks of verbal ambiguity remain. The various representational forms considered are described below, and the merits and disadvantages considered.

Structure charts

Structure charts are a familiar element of structured design methods (Bellman and Suchmann 1990). They are used to represent the relationships elicited by data flow diagrams. Their function there is to provide a method of communicating with users and sponsors about what should be in the system, and to ensure that all processes identified at the requirements stage of a system are preserved when the system is implemented, and also to enable refinement of the system described in the diagram via the application of a set of principles and rules, for instance keeping module sizes as small as possible.

They are easily understood by stakeholders. They can easily be levelled, with overview diagrams and more detailed diagrams focusing on specific areas, if a document is too complex or large for it to be represented in one chart. They are open to analysis and review: mistakes and problems can be identified by applying 'walk-through' techniques.

Data diagrams

These are used to represent the flow of data in a system. They are typically used as a requirements-gathering tool, providing data which can be translated into structure charts. As described above, they proved useful in representing the activities involved in hypertext development (Perlman 1989). Their use in hypertext design itself is questionable: the absence of data flows make some aspects of them redundant.

Flow charts

Flow charts are a method of representation popular amongst programmers at one time, but superseded by other techniques. They have more recently been adopted by CAL designers as a way of representing sequences and interactions in CAL presentations. They are useful where there is a distinct flow of information to consider, and are therefore perhaps more useful in multimedia applications. They can be adapted for hypertext design use quite readily (Berk and Devlin 1991) but are probably more appropriate for applications in which the flow of control is less in the hands of the user. In a hypertext document, the structure is in a sense passive, with control in the user's hands.

A number of flowcharting methods can be identified. Andriole (1989) quotes Van Duyn as suggesting a number, including the following:

- Conceptual flowcharts
- General system flowcharts
- Functional flowcharts
- Logic flowcharts
- Job Step flowcharts
- Work flowcharts

Structure diagrams

Structure diagrams represents a merging of the two techniques described, and are commonly used in hypertext design, although rarely overtly described as such. They simply diagrammatically represent the node and link structure of the hypertext document. They can easily be levelled for the representation of complex documents.

Storyboards

Storyboards are a technique adapted from film and TV production, and are suggested for computer interface design (Andriole 1989) and for hypertext design (Berk and Devlin 1991). They are a useful way of developing ideas for node content and screen display, and are an amenable way of displaying those ideas to users and other stakeholders. They are capable of easy development from first sketches to complex and detailed drawings showing details of text, illustrations and linking and access methods. Their principal disadvantage is that they impose a screen-based way of thinking that may have a constraining effect. It is also difficult to arrive at useful conventions for dealing with cross-screen multimedia events.

However, they are useful to the hypertext designer because they represent a level of detail that other forms of model cannot, and they represent the system in a way that users can easily apprehend.

Storyboards may correspond to individual nodes, or to screen displays within nodes, depending on the conventions used and the capabilities of the system used to produce the document. If storyboards represent nodes, the position is simple; if not then they must be typed in order to give a clear presentation as to the kind of information and its relationship to the overall node structure. So for instance a graphic shown as part of a node must be clearly identified as such, and the manner in which it is accessed and presented must be shown.

Storyboard techniques

Storyboards are typically paper-based. They may be actual screen size, or scaled down versions. They may be more or less detailed. Two forms of storyboard were tried whilst working on the electronic prospectus: blank forms, and forms with a grid based on standard screen size. The problem with the latter is that for them to be most useful they need to have a grid that is translatable into whatever grids are available for the software concerned. If the software used does not have choice of grid sizes, or even has no facility to produce a grid, like HyperCard, then the storyboard grid is not particularly useful. Storyboards may be represented via the computer. In some cases it is possible to use the same computer tools that are used for the hypertext document to produce storyboard print-outs, and these may then form the basis of early prototypes. This gets round the grid problem, in that a grid may be generated for the screen and printed out to paper.

Conclusions

In the end, two things are paramount in selecting graphical representations. Firstly, what form is the data to be included? Secondly, what form of representation is most useful to the designer? In other words, one works forward to the representation from the information or one works backwards from the desired information presentation.

An integrated approach is likely to be most efficient. The ideal situation is one in which the information required is presented in such a form as to be directly convertible into hypertext.

7.2.3.1.4 *Tools*

Tools in this context generally refers to software-based support for design (Avison and Fitzgerald 1988). This is an interesting area, and a number of hypertext design support tools have been suggested and produced (Perez 1991, Diaz and Isakovitz 1995, Johnson 1995). Some of these are already familiar, and may be included as part of proprietary hypertext software, including text editors, graphics tools and spell checkers. More specific tools include tools for cataloguing nodes and links, annotating the hypertext document, chunking material and splitting existing files, and link-checking and editing utilities. These are considered as being outside the scope of the present study as far as their construction is concerned, and are referred to generically in the design methodology at points where their use may be considered, with no specific tools being included in the methodology. Some further comments on this topic are made in the concluding chapter.

7.2.3.1.5 *Guidelines and principles*

The strengths and weaknesses of guidelines and principles have already been discussed in Chapter 4. Sometimes they become apparently set in tablets of stone over time, potentially risky in a fast-changing field. Sometimes designers fail to tailor them to suit their own working context. Sometimes they are simply wrong, as for instance Rivlin et al's recommendation of green and red as a colour combination for reading off screens - fine except for the significant minority of the population suffering from red-green colour blindness (Rivlin et al. 1990 in Dillon 1990). It was a central concern of this study to present a set of guidelines that could provide assistance both with the hypertext product and with the process of designing the product. Guidelines for the design procedure are less subject to criticism than those devoted to the product of design, and it is possible to envisage such design guidance incorporated into a design methodology.

7.2.3.1.6 *Pro formas and templates*

Methodologies typically employ a range of documents designed to ensure full and consistent documentation. In some cases, for instance SSADM, this is extensive and elaborate. Documents have to be completed and signed off when each stage and step is completed. A number of documentary support items were considered for the design methodology. These included specification forms, resource and topic lists, link record forms and other similar items.

7.2.3.2 The existing versions: a reconsideration

What can be learnt from previous efforts at hypertext design methodologies? The existing hypertext design methodologies introduced earlier - as opposed to other accounts of the hypertext design process - are based on database design precepts. Many of these methods are not described in the literature in sufficient detail. Techniques are not described in full, and there are no documentation aids or other tools included. In fairness it should be said that the only widespread accounts of these methods occur in published papers, which quite probably do not represent the methods completely, but there is no indication that these methods are any more than outlines of procedures. They also have good and useful features. They all tend to consist of general descriptions of a set of procedures, with a diagram showing the relationship of these procedures to each other. There are varying amounts of detail in the description of each procedure. Given these limitations, what might be learned from them for a design methodology?

Perlman (1989): an asynchronous methodology

This is quite detailed in overall terms, although apart from the use of data flow diagrams it does not outline any particular tools. Having originally been intended as a method of development for hypertext research, it assumes that development is an ongoing process and is based on a highly iterative model (hence the term 'asynchronous' in the title) that imposes no sequence on any of the procedures used. This means that it is possible, at least in theory, to start at any point in the cycle. This is appealing, although it sounds rather odd. It means that it is possible to intervene in an existing situation and to immediately integrate one's efforts with that existing situation, and it makes it easy to reiterate at the end of the project, which is very appropriate with for ongoing hypertext documents that require constant or frequent revision. However, this procedural model makes it hard to know where to start and when to stop. It is also largely high-level: in the end it offers little specific practical guidance at the lower levels.

Shneiderman (1989)

Rather like Horton (1990), Waterworth (1992) and Martin (1990) in that it is not quite a methodology, but a set of procedural steps which outline some significant tasks. It is useful in that it identifies some tasks missing from other accounts, and is readily accessible to those without a computing background.

Martin (1990)

This is a descriptive account of the hypertext design process which concentrates almost exclusively on the first part of the design process, especially feasibility. It contains some familiar but useful advice, including extensive detail of the minutae of hypertext design.

Van Vliet and Wilson (1993)

This has never been cited to the author's knowledge. It uses a database design orientation and gives a fairly thin account of a set of design procedures but does point out some significant tasks for the designer. It does not deal with the subject of the user's task, and uses an outdated and inadequate division between sequential and non-sequential information.

Waterworth (1993)

A somewhat unorthodox collection of procedural steps aimed at multimedia design.

Horton (1994)

This is taken from a very comprehensive book on online documentation, which includes much material belonging to the guidelines section. Horton loosely defines an overall design procedure as six stages, three top-down initial design stages, followed by three bottom-up stages.

Relationship Management Methodology (RMM) (Isakovitz et al. 1995)

This is the most recent example considered, having appeared during the course of this work. It is also one of the most detailed, and demonstrates how entity-relationship techniques might be used to specify the structure of information in a hypertext document. This parallels the problem of conversion from logical structure to physical structure found in information systems design, and entity-relationship diagrams are a technique adopted from this field. Its most interesting feature is the way that the original entity-relationship diagram is worked up into a hypertext structure diagram via an intermediate stage. This parallels the less formal process of generating a structure diagram used by the author in designing the electronic prospectus.

Hypermedia Design Model (HDM) and Object-Oriented Hypermedia Design Method (OOHDM) (Garzotto 1993, Shwabe and Rossi 1995)

HDM is described by its authors as a design model rather than a design methodology, but in many respects it shares characteristics with design methodologies. Its separate nature as a model is however shown by its application in describing hypertext documents authored with no knowledge of the model itself (Garzotto et al 1995). In its more recent development, Object-Oriented Hypermedia Design Method, it is more clearly a design methodology. It describes four steps in hypertext design:

1. Domain analysis.
2. Navigational design.
3. Abstract interface design.
4. Implementation.

It uses an extensive technical vocabulary that is hard for the non-computer scientist to understand and apply - certainly in the author's experience.

Whitby (1993)

Also examined was the method described by Whitby (1993) for the production of paper prospectuses. This provided a closer connection with the original subject area of this study, and also provides a reference point from the publishing world. The method of representation used here was a flow-chart method. It describes a number of tasks, most of which are generic publishing tasks, but includes a number of tasks which are specific to the prospectus design situation.

7.2.3.3 What can be learned from design methodologies to date

It is doubtful if there is a right or even a most useful design methodology. As Avison and Fitzgerald (1988) suggest, pursuit of perfection is probably a hopeless cause. Even the concept of a most useful methodology is difficult, hampered by difficulties in definition in the first instance, and then question of context: useful for what? All the methodologies and procedures mentioned in this thesis have their useful aspects. If anything they have a general tendency to be strongest on the early aspects of design, less clear about what happens in the middle sections, and a little more clear about the final stages. This is reasonable in that it appears that the early analysis stages have a crucial influence on what happens next (Bellin and Suchman 1990), and it is understandable in that the middle stages of design are the ones of maximum insecurity (often coinciding, as Horton (1994) describes, with a morale drop). But this approach leaves a hole roughly where the conversion from logical to physical design occurs, that is, where the information structure is translated into hypertext. One of the author's concerns has been to represent the information structure in such a way that it can easily be converted into a hypertext structure.

This reflects the author's perceptions about hypertext design: that it moves from fairly clear-cut initial premises to an amorphous middle section where a kind of chrysalis-like process takes place. The product exists in a state of chaos for a period, during which it appears that no progress is being made and that no final artefact is likely to emerge. Yet quite suddenly a working product emerges. This phenomenon seems to be based on the problems inherent in translating the logical structure of the required information into a navigable physical structure, in particular the lack of any perceivable structure in the designed artefact until a substantial number of links are in place. This suggests that Horton's initial top-down method followed by a modular bottom-up approach may well be useful. It means a clear overall structure from very early on, followed by completed sections which can be slotted into the structure. Even better, a methodology that supports the move from top-down to bottom-up and back again.

7.2.3.4 How detailed and prescriptive should a methodology be?

In the information systems design world, methodologies are often highly detailed and constrained. They offer a complete package, down to providing forms and charts to guide the design process. This can be highly bureaucratised, as for instance is SSADM. Some include software tools as well. This author considers that such an approach is unlikely to be appropriate to the hypertext context for several reasons. Firstly, the range of potential hypertext authors involved is too great. The designer of an information systems design methodology can assume certain qualities in the potential users of that methodology. They form a relatively homogenous group. At a minimum, systems designers can be expected to have a shared professional background and some common skills in such areas as data diagrams. It is therefore possible to make certain assumptions, and base the design of a methodology upon those assumptions. (It should be noted that even in this relatively favourable situation, methodologies are still known to be flawed - see Jayaratna's (1994) criticisms mentioned in the literature review, Chapter 2.) This favourable situation does not apply in the case of hypertext development in general, and this is no less the case where the development of hypertext electronic prospectuses are concerned. The literature shows that people with a wide range of backgrounds are involved. This leads to the possibility that a methodology that is agreeable to the predispositions of all of these people may not be possible or appropriate.

Next, the design context is different. Hypertext documents, especially the smaller type under consideration here, are rarely concerned with critical data flows at the heart of organisations, and they are at this scale less likely to involve such a large number of people. This makes the organisational aspects of hypertext design more manageable, and reduces the necessity for a highly bureaucratised method.

7.2.3.5 A rationale

Having considered the relevant domains, this attempt to find a hypertext methodology follows the argument of Avison and Fitzgerald (1988) and others that methodologies should be contingent rather than prescriptive in the light of the great variation in analysts/designers and situations that exists in the world. This parallels the arguments discussed earlier regarding design guidance from the HCI perspective by Smith (1988) and others, which suggest that such guidance must be tailored to situations and contexts. Hence the proposed design methodology presented here emphasises flexibility. A possible model for this is Avison and Wood-Harper's (1990) Multiview methodology, which adopts techniques from different methodologies at different stages and for different contexts, and attempts to provide some room for individual preference and situational variation.

7.3. Method

Before going on to examine the procedure followed in creating the design methodology, it is necessary to consider the role of the earlier work and the resulting data. The question is how might a design methodology be constructed from the data gathered via the taxonomy, the electronic prospectus and the case study?

Beyond that, what reasoning process is appropriate? The endeavour can be seen as falling within the usual paradigm of scientific endeavour. It is a matter in the first instance of generalising from the data gathered. Certain things may be said which generalise to the relevant design situation. Certain of the principles and guidelines that have been proposed seem to have sufficient weight to generalise to all such situations. The experiences of those studied in the case studies suggest more generalisations. These generalisations then need to be assembled into a framework for design. The best possible way of doing this was considered to be to follow an iterative process just as a designer of other artefacts might do, but where such a designer would be in the position to test the resulting artefact with users, it was necessary to reflect on the result with regard to the data gathered and the internal logic of the procedures suggested.

7.3.1 THE DATA

7.3.1.1 The design guidance

The design methodologies are considered above. A wealth of other design guidance was gathered and classified, as described in Chapter 4. This was used as a source for tasks and other elements not mentioned at methodology level.

7.3.1.2 The electronic prospectus

The electronic prospectus provided less firm data, but rather a perspective or orientation based on the preunderstanding obtained in the process.

7.3.1.3 The case study

The case study suggested the following recommendations:

Unless extensive in-house programming support is available and a choice to produce a custom-built document is taken, software should be carefully investigated, and a fully informed choice made. This should involve a systematic procedure for evaluation, preferably based on a clear requirements specification. There are no easy solutions here though. It is not enough to say that the software should not impede the design solutions chosen. For the time being they will. The point is to minimise such effects.

Video should only be used if it provides genuine added value as its current technical imperfections are otherwise likely to cause problems out of proportion to the likely return.

Design teams should either include members with a range of software skills or should have a clear programme for the acquisition of such skills, including a strategy for an appropriate division of labour.

Close attention should be paid to version control, with a systematic record of versions being maintained.

7.3.2 PROCEDURE

A top-down approach was taken initially. This is a common approach in systems design generally, and has the advantage of enabling complex tasks to be rendered manageable by a process of decomposition into simpler units. This meant first examining the various hypertext authoring tasks as described in the various methodologies examined earlier at the highest level and producing a set of stages. Data from the methodologies were then aggregated under the appropriate procedural steps, effectively a bottom-up process, using a series of working tables. The procedural steps were assessed at this stage and modified. The most difficult section concerned the information analysis and design stages, where apparent overlaps between Information Design and Information creation and input, explained by the fact that the former is composed of high level tasks, and the latter of low-level tasks, made it hard to separate out the two stages. A final 'Post-design' phase was also added at this point. The contents of the stages were reorganised to fit this revised structure, and the whole was examined for weaknesses such as missing tasks. The guidelines taxonomy was checked as part of this process and a revised table drawn up. This was the basis for the staged methodology shown in M01 (Figure 7.1, p205) Each of the stages was then isolated as a series of stage documents, MT1-11 (Appendix 4).

However, there was a problem of representation here. It was apparent that this table was naturally viewed as a set of linear procedures, and ran the risk of being used in a linear fashion. This was misleading. It is clear that what designers do involves a high degree of iteration, as shown both in the case study evidence, and other work (eg. Hix and Hartson 1989) and it was, as previously noted, part of the design philosophy behind the methodology to support the ways in which designers tend and prefer to work. Although some iteration was referred to on the diagram, it was not clear how important this was. It was also not clear that certain tasks, such as task analysis and user analysis could be carried out concurrently, and that other tasks such as user interface design, could be carried out relatively early in the design cycle, and may even benefit from this.

So an alternative structure and representation was sought. One approach which suggested itself was the data flow diagram (DFD) approach used in Perlman's (1989) asynchronous methodology. Although the methodology itself had its weaknesses, the DFD approach to representing the design process was attractive. Data flow diagrams are useful for the representation of process. They model the process concerned by dividing elements of the process into processes, data flows and data stores. Each of the stages in the design methodology can be seen as a process in its own right, containing a number of tasks. Data can be seen as the outcome of many of these processes, data on the task, data on the users, data from evaluations, and so on. These outputs form the inputs to other processes. These data flows are symbolised by arrows. Some data is stored, for instance the requirements specification is a data store. Finally, a semi-asynchronous version was produced, using a data-flow diagram representation, diagram M02 (Figure 7.2, p206). This gives the user of the design methodology two options: to follow the staged version sequentially or take a more flexible approach based on the asynchronous approach. This latter is more realistic, but the advantage of a sequential staged approach is that it is easier in a highly modular framework to retain control of tasks and to finish them.

7.4. The results: The proposed design methodology

The finished methodology is included as Appendix 4.

7.4.1 THE PROCEDURAL PHASES IN DETAIL.

The methodology consists of eleven procedural phases. These are Feasibility, Stakeholder analysis, Task analysis, Requirements analysis and specification, Information content analysis, Information design, Information content creation and input, Linking and access, User interface design, Testing and evaluation, and Post-design. They are shown below (Figure 7.1 p205) as a staged procedure, which it is suggested some users of the methodology may prefer to follow as it offers maximum modularisation.

However, the procedure may also be followed in a semi-asynchronous manner, as indicated in Figure 7.2 (p206). This diagram shows a data-flow representation of the hypertext design task. The ellipses represent activities, whilst the parallel lines represent data stores. Some inputs from outside the system are shown, but not all, in the interests of clarity.

Although initially inspired by Perlman's asynchronous methodology, this methodology assumes that, unlike Perlman's methodology, there is an approximate starting point in feasibility. It also shows two halves to the design procedure, roughly corresponding to analysis and design.

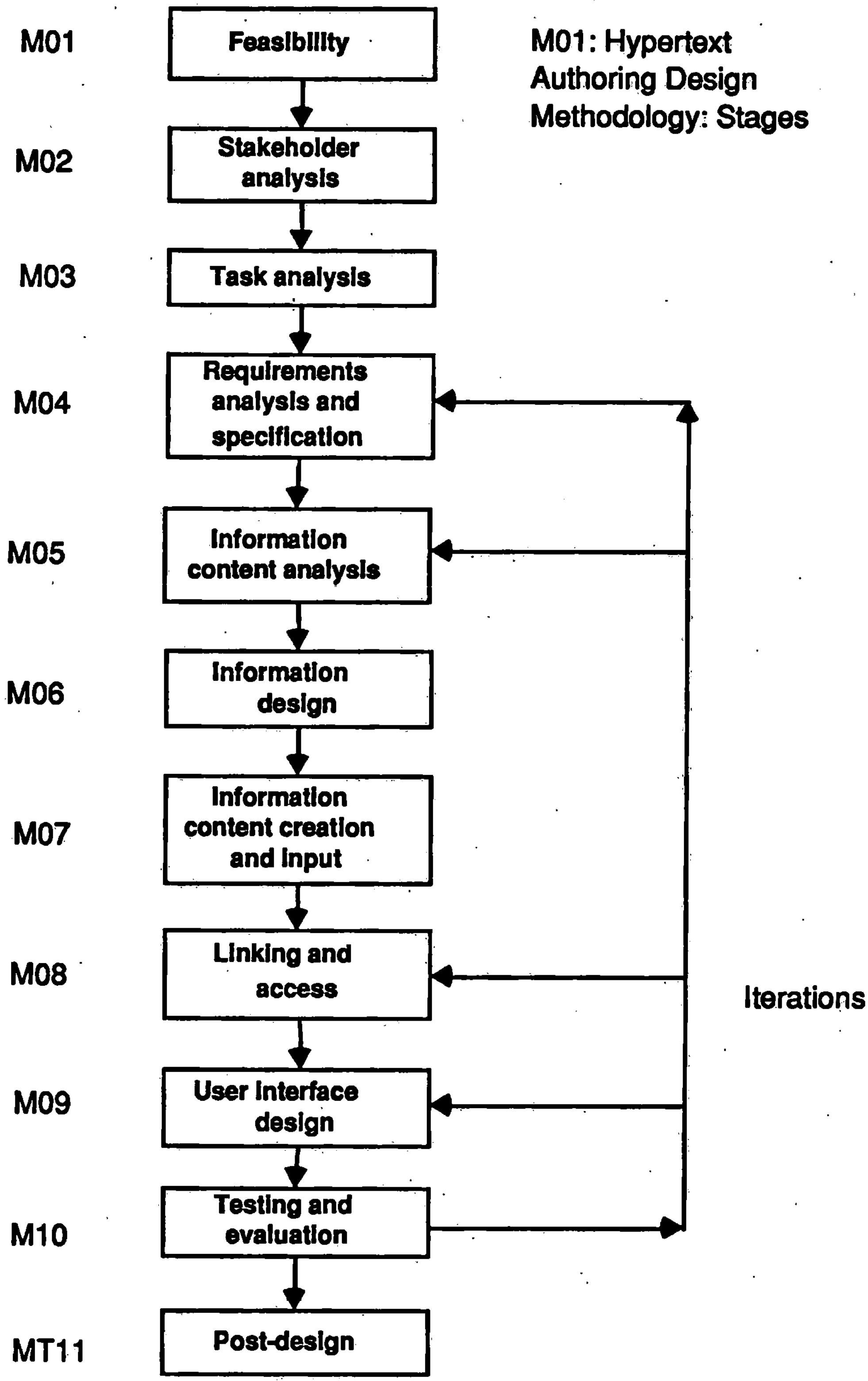
The full design methodology is included as Appendix 4. Each of the stages in the diagram (Figure 7.1) overleaf has a related document (MT1-11), with advice for use, plus other elements, usually forms, as indicated.

7.4.1.1 Feasibility

The feasibility part consists of a check-list of items for consideration. It also includes a software evaluation checklist.

7.4.1.2 Stakeholder analysis

Stakeholder analysis identifies all those with a significant interest in the document, and a pro forma is included for listing the various stakeholders identified. Designers are encouraged to develop scenarios of stakeholders' use of, and encounters with, the document, to 'define and develop a sense of user space.' (Tognazzini 1992 p74). The users are the most significant single sub-group amongst the stakeholders, and a separate form is reserved for a user analysis.



(Figure 7.1) M01: Hypertext Authoring Design Methodology: Stages

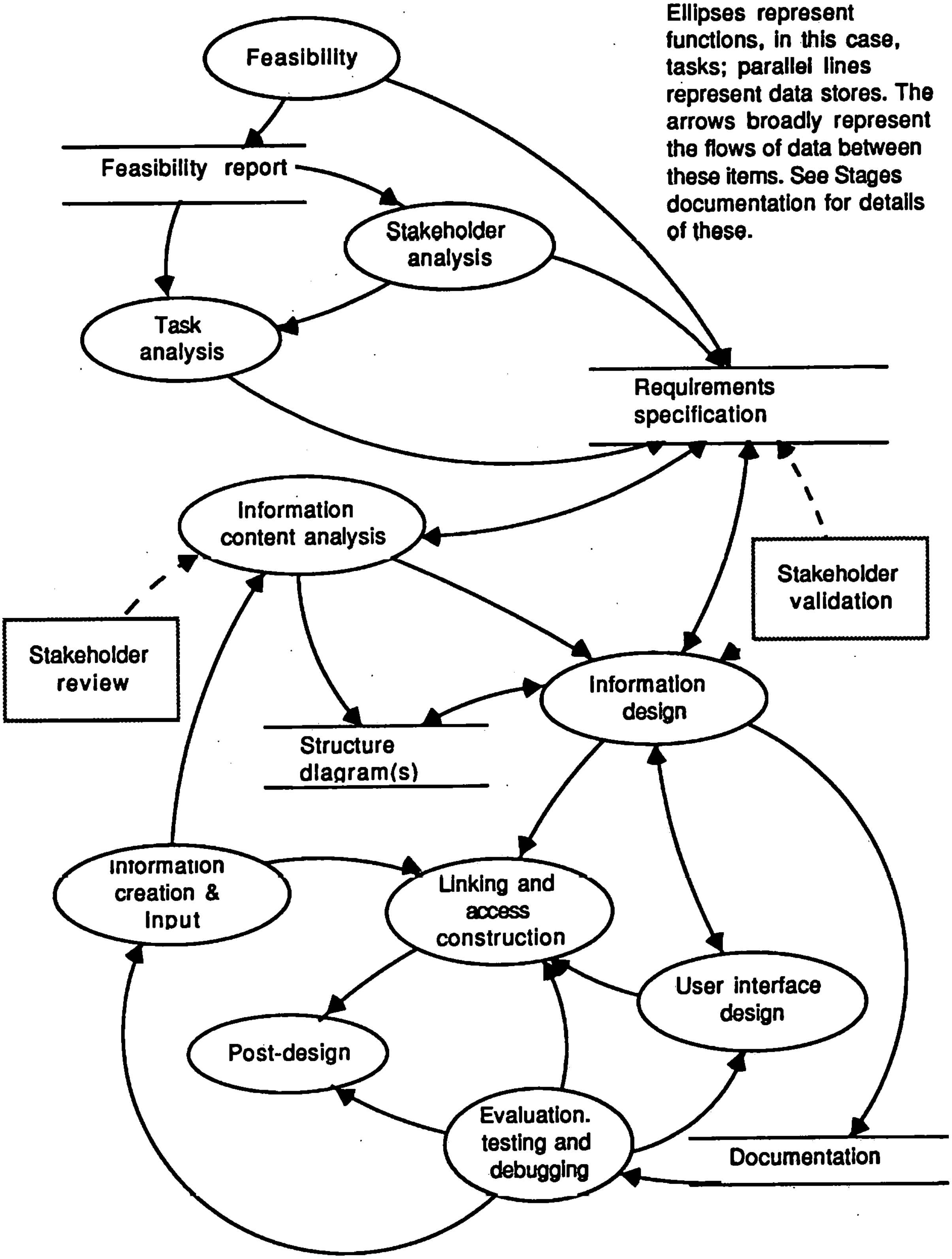


Figure 7.2 M02: Hypertext Authoring and Design Methodology: a data-flow diagram representation.

7.4.1.3 Task analysis

This section proposes a general approach to task analysis for hypertext, as opposed to adopting a specific method, and lists a selection of appropriate techniques and a report form.

7.4.1.4 Requirements analysis and specification

This is seen as a process of synthesis from the above two stages, resulting in a requirements specification document, for which a template is provided. Requirements for hypertext follow a conventional model in dividing up into functional and usability or user interface requirements, but where it is normal to consider data requirements, in this context it seems more appropriate to use the term 'information requirements', following the use of Isakovitz et al. (1995) and others.

In the staged version of the methodology (M01), the requirements specification is produced relatively soon in the design process. The need to involve users, and the relative unfamiliarity of the designer with the design situation in the early phases of analysis will probably mean that the requirements change over the design process.

7.4.1.5 Information content analysis

This identifies tasks for this stage and focuses on techniques for the identification of the information content to be included and the elicitation of its structure, including specifying topic areas or node clusters and identifying media requirements. This stage is a rather awkward one to define: the rationale here is that the information content is obviously a key element of hypertext documents providing information, and is therefore in need of some form of analysis before design work can proceed. However, the further boundary of this stage is hard to define. It could be argued that ideally there should be no distinction where the information for a hypertext document is concerned: that there should be a direct translation between information structure and the hypertext structure that will represent it. But the reality is rarely this transparent. Given a specific body of information, a range of options is likely to exist as to how to structure the information. For instance, Horton (1994) describes a number of ways of ordering information: in sequence, in hierarchy, in grids and in webs. Looking at sequences alone he distinguishes a number of ways of ordering sequential information: spatial order, chronological order, logical order, associational order, climactic order, anti-climactic order, clincher order and indexed order. It is quite likely that a body of information may be capable of structuring and ordering in more than one way.

In this context, methods of eliciting the information requirement in a way which presents a coherent and ordered structure are attractive.

7.4.1.6 Information design

This is a little difficult to separate out from the previous phase, one reason why an asynchronous approach can have advantages. This part of the design methodology presents a number of techniques which may be chosen for the translation of the information structure elicited by the information content analysis into a physical hypertext design. This includes hypertext structuring and typographical structuring.

7.4.1.7 Information content creation and input

This is best considered as a bottom-up task. How exactly it is orderod will depend on the earlier decision to use existing materials or author from scratch. This section contains a checklist of nodes based on the earlier storyboards. This checklist includes the node reference, the original file-name of the material, the media type, the file format, date created, author, date of completion.

7.4.1.8 Linking and access

This aspect has been treated in this methodology as a separate stage, although it shares certain characteristics and concerns with both information design and user interface design. The main feature here is a record sheet for recording the correct creation and completion of links, including link source, destination, type and whether completed. This reflects the common experience of designers failing to ensure all links are completed, and functioning in the way that their representations suggest they should. The main problem here is keeping things simple enough to encourage use: if one imagines a document of 100 screens with an average of six links per screen, there will be 600 links to record and monitor. Some authoring systems will produce a list of links and details as to their status, perhaps with a little scripting, and such an approach would be preferable.

7.4.1.9 User Interface design

This concerns the dialogue design, information display, user action and on-line assistance. User interface design should be informed in the first place by the user interface requirements elicited earlier.

The main elements here are a list of concerns, based on user interface guidelines for hypertext. This is an attempt at a non-prescriptive approach to guidelines that may be adapted by the designer as appropriate. Once again, user interface design is not as inseparable from other stages as this staged process model can imply. Information structuring impinges upon interface design because it takes place at a number of levels including screen level, in terms of how the information is ordered on screen (Tullis 1988, in Preece et al 1994).

The user interface design stage should result in a list of usability items which can form the basis of evaluation.

7.4.1.10 Testing and evaluation

There are a number of questions to consider here. Beaulieu and Mellor (1995) sum these up as:

When is it appropriate to evaluate?

- during the prototyping design phase or after implementation?

What should be evaluated?

- the system?
- user behaviour?
- information content?
- educational goals?

Which criteria or measures should be used to evaluate?

- usage statistics
- ease of use?
- user satisfaction?
- amount of information retrieved?
- time to retrieve a result?
- degree of navigation?

Which data collection methods are appropriate?

- questionnaires/interviews?
- talk-aloud?
- observation?
- transaction logging?

This is a by no means exclusive summary, but gives some indication of the range of possibilities. For this methodology, the concern with small to medium sized documents suggests that evaluation should be relatively constrained, and that techniques should be as cheap and quick as possible whilst still retaining efficiency. Inspiration in this regard comes from Nielsen's (1989) notion of discount usability, a set of techniques for evaluation that is designed to offer a quick and cheap way of incorporating evaluation into the design process using a mixture of user participation and heuristics.

Key points for evaluation appear to be after requirements have been established, when the information structure is established, when a usable interface is established, and when the system is roughly finished. More evaluation is perhaps better, but takes time.

On the question of 'what to evaluate', the most important aspects are user satisfaction with the system in general and with the information content and its accessibility. This being the case, the key criteria are ease of use, user satisfaction in terms of the amount - and quality - of information retrieved and the time to retrieve a result.

7.4.1.11 Post-design

The post-design phase covers a group of miscellaneous tasks that are often unconsidered in methodologies, such as user documentation, planning for updating, and backup routines

7.4.1.12 Instructions for use of the design methodology

Some explanatory and instructional text accompanies the methodology. There is a general introduction accompanying the overview diagrams, and each of the items for the various stages - MT1-11 - contains instructions for that stage.

7.4.1.13 Guideline sets

Several sets of design guidelines are included, together with advice for use. These are guidelines for nodes, guidelines for links, guidelines for structure, and user interface guidelines. These were extracted from the guidelines database. They are presented as concerns as a precaution against slavish adherence to the guidelines.

7.4.1.14 Testing and validation of the Design Methodology

To date it has not been possible to try out the full methodology on an authentic and full-scale project. This remains the next aim at the time of writing.

A number of the elements have been tested by the author as part of the electronic prospectus process, but it is clearly difficult to obtain valid information about the methodology as a whole through the author's own efforts.

8: Conclusions and Future Developments

8.1 Introduction

This chapter examines the contribution to knowledge of the research, and makes some suggestions for further useful research in the area described.

8.2 Contribution to knowledge

This research contributes to knowledge in three areas: a new design methodology, a taxonomy of design guidance, and a case study of hypertext electronic prospectus design.

8.2.1 Design methodology

The primary contribution to knowledge is the Design Methodology itself.

This is an integrated and systematic approach to hypertext design which contains detailed support for the designer, and provides a structured methodology particularly suitable for application by designers without a background in computing science, information systems or database design. It is based on a synthetic approach, combining advice about hypertext design procedures from a number of sources, including where appropriate the existing design methodologies.

8.2.2 Taxonomy

This collation and classification of the existing design guidance constitutes a contribution to knowledge by placing together much of the design guidance material in an integrated and categorised form.

8.2.3 Case study

The case study adds to the body of knowledge about hypertext designers at work. It provides a detailed examination from a retrospective perspective of the design process for smaller hypertext documents. It also demonstrated a case study methodology specifically adapted for retrospective inquiry into such design situations, and utilised a case study protocol which can be used for further comparative case studies in the area of hypertext design. It shows how Yin's (1994) concept of a chain of evidence can be applied to software design processes in order to answer questions about those processes.

8.2.4 Relationship to the literature

The literature review identified a number of gaps in existing knowledge. Firstly it was felt that although the hypertext design process had been reported in a number of places, often on the basis of first-hand experience, very few of these represented systematic attempts to study the process. Secondly, support for hypertext designers tended to focus on the quality of the resulting product, especially in terms of its usability. This was reflected by the hypertext design guidelines to date, which tended to emphasise the quality of the resulting product rather than the design process, as exemplified by guidelines sets like those of Hardman and Sharratt (1990), and could also be seen in the orientation of the existing design methodologies such as OOHDM (Schwabe and Rossi 1995) and RMM (Isakowitz et al. 1995). Such design methodologies for hypertext as these were unsatisfactory in that they were limited in their perspective, being heavily influenced by concepts from software engineering and database design, and in their level of detail. The problems posed for the designer in the process of designing effective hypertext were relatively neglected. There were few texts which drew together useful design techniques in an easily applicable manner.

8.2.4.1 The design methodology

As outlined in the literature review, when the author first investigated this subject, there was only one model applicable to hypertext application design (as opposed to data models such as the Dexter model (Halasz and Schwartz 1990)) applicable to hypertext system development) in existence, Garzotto et al.'s (1993) Hypertext Design Model (HDM). This was developed into what the authors regard as a design method during the course of this research, Object Oriented Hypertext Design Method (OOHDM), (Shwabe and Rossi 1995). There were only two published design methodologies intended particularly for hypertext, those of Van Vliet & Wilson (1993), and Perlman (1989). Whilst this research was in progress, a further development in design methodologies for hypertext was Isakowitz et al.'s (1995) Relationship Management Methodology (RMM). This is characterised by a clearly delineated data model which is based on entity-relationship modelling and was based on HDM and its successor, HDM2.

The most fully realised of these design methodologies such as RMM employ formal methods adapted from software engineering and database design to represent the products of each design stage. It was considered that such techniques require at least some specialised knowledge, being intended in their original role for use by trained analysts. For the author whose primary purpose is to design a hypertext document this may not be appropriate or helpful. It was felt that there was therefore room for a design methodology which did not use formal methods, but which provided some clear guidance as to what tasks were appropriate when.

There were also a number of descriptions of the total hypertext design process which have something in common with design methodologies, such as those described above by Shneiderman (1989) and Martin (1990), but these also tend to be short of detail. However, they did have the advantage of being aimed at non-specialists, and expressed in clear non-technical language. Hence the rationale behind these attempts to make a design methodology for hypertext was essentially to fill the gap between the specialist methodologies and such accessible but undetailed accounts as those of Shneiderman and Martin.

In this respect, the design methodology appears to be a useful contribution to the area, in that it is clearly expressed in non-technical language, yet still supplies detailed support in areas that need it such as the structuring of information. Hence it falls between the technical aspect of methodologies such as OOHDM and RMM, and the simple lists of Shneiderman, Martin and others.

8.2.4.2 The taxonomy

Existing design guidance for hypertext appeared from the literature to be somewhat chaotic, with much overlap. There was an apparent tendency for guidance to be oriented towards the document rather than the design process, and guideline sets did not always separate the two types. To what extent this was true was not entirely clear. This in itself prompted the notion that some consolidation would be useful. Earlier precursors of such consolidatory work exist, notably Hardman and Sharratt (1990), who collated and edited general user interface design guidelines to identify a set specific to hypertext, but they tend to be restricted to principles and guidelines for designing the features of the document. This taxonomy clearly identifies and separates (with the exception of situations where it is impossible to apply a rigid distinction) guidelines about features from guidelines about designing.

8.2.4.3 The case study

Although there is a considerable body of knowledge about various aspects of hypertext, studies of the hypertext design process have been uncommon. Most work has tended to concentrate on the activities of software designers in general, such as Guindon (1977, 1988, 1989). The work of Nicol (1988) is a rare example, but limited itself to designers using HyperCard. Most case studies in the literature are self-reported and tend to be oriented towards representing particular aspects of the design problem, such as reports of the SuperBook design process (Egan et al. 1989), which concentrates on the iterative method used to make the system better rather than the design process for the designers. The case study takes a detailed retrospective perspective on the design team studied, reducing the opportunity for the researcher to influence the design process. It presents an account of the design process based on a clearly described chain of evidence, and manages to successfully answer some key research questions.

8.3 Further research

8.3.1 IMPROVING AND EXTENDING THE DESIGN METHODOLOGY THROUGH TESTING AND EVALUATION

It is already apparent to the author that there are aspects of the methodology which are open to improvement. In particular, the methodology does not yet provide its own integrated techniques for such stages as task analysis and evaluation but instead suggests the designer choose from a range of existing techniques. It is not necessary to consider entirely new techniques here, but rather to ensure that techniques integrate closely with the general design approach taken by the design methodology.

The best way to identify and make these improvements now that the design methodology is completed is to apply the techniques of iterative design to the design methodology. The methodology therefore requires evaluation in the first instance.

A number of evaluatory approaches would be appropriate. The most obvious would be the application of the design methodology in practice, preferably by the kinds of novice and intermediate designers most likely to benefit from it. This need not involve the design of entire documents using the complete methodology, although this would be an optimum approach. Stages and techniques from the methodology could be evaluated and improved on an individual basis.

An alternative would be an action research approach where the design methodology was applied by the researcher to a real hypertext design situation, and the effectiveness of the intervention evaluated.

Also useful would be validation by a designer or designers with experience in electronic prospectuses or similar information systems.

Another aspect is the use of software tools during the hypertext design process. This aspect was not within the scope of this research, but is one that should be considered in order to maximise the value of the methodology to designers. As noted previously, a number of tools for supporting hypertext design already exist (Perez 1991, Diaz and

Isakovitz 1995). Some of these are recommended in the design methodology, and it is relatively simple for the designer to add stand-alone tools as required, but the next step is to provide an integrated environment to support the use of the design methodology, on the basis of such tools as analyst's workbenches, used in information systems analysis (Longworth and Nicholls 1987) and Computer-Aided Software Engineering (CASE) tools. This approach has already been adopted by Diaz and Isakovitz (1995) for RMM, with their RMCase tools, and Nanard and Nanard (1995).

8.3.2 EXTENDING AND RECLASSIFYING THE TAXONOMY

The taxonomy would benefit from extension to provide further details of the background to the production of each set of guidelines and each design methodology. This means including all references made by the original authors and providing more details as to context and purpose. An ongoing updating process is also desirable.

8.3.3 FURTHER CASE STUDIES

This study has confined itself to a fairly narrow domain, that of electronic prospectuses. It would be useful to examine hypertext authoring in other domains. It would also be informative for the future researcher to get a closer look at what happens in real hypertext design situations. The case study methodology applied after the event is useful in that it means that evidence can be obtained which is less likely to be tainted by the researcher's intervention, but does not allow for observation of design situations and problems as they happen. It is arguable that this would provide greater insight into the kinds of methods and techniques that would provide the best support in this situation. A participant observation or action research approach could provide useful evidence here. Alternatively a task analysis approach could be used, employing a recognised task analysis method such as Hierarchical Task Analysis (Annett and Duncan 1971).

8.3.4 DEVELOPMENT OF THE CASE STUDY PROTOCOL

The case study protocol has the potential to become a standardised instrument for inquiry into hypertext design processes. For this to be effective, the protocol requires revision and extension. In particular it needs a reconsideration of the interview process, perhaps by augmenting it with a standard questionnaire, in order to ensure that information is not lost in the process of a semi-structured interview. It should also adopt a procedure for analysis of thc existing hypertext documents found, to record and analyse their structure, and to allow for 'reverse engineering' and similar examinations of the documents (Dillon and McKnight 1995).

Robert Urquhart Kemp, BA, MSc.

Developing a design methodology for the construction of hypertext and hypermedia, with particular reference to hypertext electronic prospectuses

Degree of Doctor of Philosophy

Information Science

31st May 1997

Queen Margaret College, Edinburgh

AWARDING BODY:
THE OPEN UNIVERSITY

Volume 2 of 2

Bibliography

ACM hypertext compendium. (1991). NY: ACM Press.

Akeroyd, J. (1990). Information Seeking in Online Catalogues. *The Journal of Documentation,* 46 (1) 33-52.

Akscyn, R., McCracken, D. & Yoder, E. (1987). KMS: A Distributed Hypermedia System for Managing Knowledge in Organisations. In *Proceedings of ACM Hypertext '87 Conference* Chapel Hill, NC., 13-15 November, (pp. 1-20).

Allen, B.L. (1991). Cognitive Research in Information Science: Implications for Design. *Annual Review of Information Science and Technology,* (26) 3-37.

Alschuler, L. (1989) Hand-crafted hypertext: lessons from the ACM experiment. In E. Barrett, (Ed.), *The Society of Text: Hypertext, Multimedia and the Social Construction of Information* (343-361). Cambridge, MA: MIT Press.

Alschuler, L. (1993). Special Section: Standard Generalized Markup Language -Introduction. *Technical Communication,* (Third quarter) 376-378.

Altrichter, H. (1991). Do we need an alternative method for doing alternative research? In O. Zuber-Skerritt (Ed.), *Action Research for Change and Development* (pp. 79-92). Aldershot: Avebury.

Andersen, P. (1990). Towards an aesthetics of hypertext systems. A semiotic approach. In A. Rizk, N. Strietz & J. Andre (Eds), *Hypertext: Concepts, Systems and Applications* (pp. 224-237). Cambridge: Cambridge University Press.

Anderson, B., Ashworth, G., Barnard, P., Boucherat, P., Downton, A., Johnson, H., Johnson, P., Jones, P. & Leedham, G. (1991). *Engineering the Human-Computer Interface.* London: McGraw-Hill.

Andriole, S.J. (1989). *Storyboard Prototyping: a New Approach to User Requirements Analysis.* Wellesley, MA: QED Information Sciences Inc.

Annett. J. & Duncan, K.D. (1967) Task analysis and training design. *Occupational Psychology,* (41) 211-221.

Antill, L. (1985). Selection of a Research Method. In E. Mumford, R. Hirschheim, G. Fitzgerald & T. Wood-Harper (Eds). *Research Methods in Information Systems* (pp. 203-215). Amsterdam: Elsevier Scientific Publications

Apple Computer Inc. (1989). *HyperCard Stack Design Guidelines*. Reading, MA: Addison-Wesley.

Apple Computer Inc. *Human Interface Guidelines: The Apple Desktop Interface.* Reading MA: Addison-Wesley.

Archea, J. (1988). The myth of application: rethinking the relationship between inquiry & action. Design *Research News* , 19 (1).

Arents, H. & Bogaerts, W.L. (1991). Towards an architecture for third-order hypermedia systems. *Hypermedia,* 3 (2) 133-152.

Arents, H.C. & Bogaerts, W.F.L. (1992). Hypermedia and the User: Defining Hypermedia by Its Application Areas and Its Utilization Characteristics. *Microcomputers for Information Management,* (9) 17-33.

Arents, H.C. & Bogaerts, W.F.L. (1993). Concept-Based Retrieval of Hypermedia Information: From Term Indexing to Semantic Hyperindexing. *Information Processing and Management,* (29) 373-386.

Ashlund, S., Mullet, K., Henderson, A, Hollnagel, E. & White, T. (Eds), *Bridges between worlds, INTERCHI'93 Conference Proceedings, Adjunct Proceedings*. Reading, MA: Addison-Wesley.

Avgerou, C. and Cornford, T. (1993). *Developing Information Systems: Concepts, Issues and Practice.* London: MacMillan.

Avison, D. & Wood-Harper, T. (1990) *Multiview Methodology.* Oxford: Blackwell Science Publishers.

Avison, D., Shah, H., Powell, R., & Uppal, P. (1992). Applying methodologies for information systems development. *Journal of Information Technology,* 7(3) 127-140.

Avison, D.E. & Fitzgerald, G. (1988). *Information Systems Development: Techniques and Tools* (1st ed.). Oxford: Blackwell Scientific Publications.

Avison, D.E. & Wood-Harper, A.T. (1991). Information systems development research: an exploration of ideas in practice. *The Computer Journal,* 34 (2) 98-112.

Avison, D.E. (1993). The discipline of information systems. In *Proceedings, ISTIP-91 - Fourth UK Conference on Information Systems Teaching: Improving the Practice* - Sunningdale 3-5 April 1991.

Backer, P. (1990). Authoring electronic books. *Computer Education,* (66) 2-5.

Balasubramanian, V. (1994). State of the art review on hypermedia issues and applications, WWW page:

Balasubramanian, V., Min Ma, B. & Joonhee, Y. (1995). A systematic approach to designing a WWW application. *Communications of the ACM* August, 38 (8), 47-56.

Balmer, J.M.T & Ennis, S. (1994). Corporate Identity: Is it more than meets the eye? *Marketing Education Groups Conference* , Coleraine, 4-6 July 1994.

Bannon, L. (1991). From human factors to human actors: the role of psychology and human-computer interaction studies in system design. In J. Greenbaum & M. Kyng (Eds). *Design at Work: Co-operative Design of Computer Systems* (pp. 25-44). Hillsdale, NJ: Laurence Erlbaum Associates.

Baoming, Z. (1993). Authoring a hypertext database - experiences with HyperPad. *Aslib Proceedings*, 45(1) - Jan, 19.

Barrett, E. (1989) (Ed.). *The Society of Text: Hypertext, Multimedia and the Social Construction of Information* . Cambridge, MA: MIT Press.

Barrett, E. (1992). *Sociomedia: Multimedia, Hypermedia, and the Social Construction of Knowledge*. Cambridge, MA: MIT Press.

Basalla, G. (1988). *The Evolution of Technology*. Cambridge: Cambridge University Press.

Bates, M. (1990). Where should the person stop and the information search interface start? *Information Processing and Management*, 26 (5) 575-591.

Bawden, R. (1991). Towards action research systems. In O. Zuber-Skerrit (Ed.), *Action Research for Change & Development* (pp. 10-35). Aldershot: Avebury.

Beaulieu & Mellor (1995). The Micro-Gallery: an evaluation of the hypertext system in the National Gallery, London. *New Review of Hypermedia and Multimedia*, 1, 233-260.

Bechtel, B.L. (1990). Inside Macintosh as Hypertext. In A. Rizk, N. Streitz, & J. Andre, (Eds), *Hypertext: Concepts, Systems & Applications* (pp. 312-323). Cambridge: Cambridge University Press.

Beck, A. & Zeigler, J. (1991). New Approaches in Software engineering for interactive systems. In H.J. Bullinger (Ed), *Human Aspects in Computing, Design & Use of Interactive Systems* (pp. 783-787). Amsterdam: Elsevier.

Begoray, J.A. (1990). An introduction to hypermedia issues, systems and application areas. *International Journal of Man Machine Studies*, (33) 121-147.

Bellin, D. & Suchman, S. (1990). *The Structured Systems Development Manual.* New Jersey: Yourdon Press Computing Series.

Benbasat, I., Goldstein, D.K. & Mead, M. (1987). The case research strategy in studies of information systems. *MIS Quarterly* (Sept.) 369-386.

Benest, I.D. (1991). An alternative approach to hypertext. *Education & Training Technology International,* 28(4) 341-346.

Berk, E. & Devlin, J. (1991) (Eds). *The Hypertext/Hypermedia Handbook.* NY: Intertext Publications/McGraw Hill.

Berk, E. & Devlin, J. (1991). Bringing your application together with programming or authoring. In Bunzel M.J. & Morris, S. K. (1992) (Eds). *Multimedia Applications Development Using DVI Technology* (pp 203-216). NY: McGraw Hill.

Bernstein, M. (1991). The navigation problem reconsidered. In Berk, E. & Devlin, J. (1991) (Eds). *The Hypertext/Hypermedia Handbook* (pp. 285-297). NY Intertext Publications/McGraw Hill.

Bieber, M.P. & Kimbrough, S.O. (1992). On generalizing the concept of hypertext. *MIS Quarterly,* 16(1) 77-93.

Bolter, J. (1991). *Writing space: The Computer, Hypertext, and the History of Writing.* North Carolina: Lawrence Erlbaum Associates Inc.

Booth, P. (1989). *An Introduction to Human-Computer Interaction.* Hove: Lawrence Erlbaum .

Boyle, C. & Ratliff, K. (1992). A Survey & Classification of Hypertext Documentation Systems. *IEEE transactions on professional communication,* 35(2) 98-111.

Brockmann, R. J., Horton, W.R. and Brock, K. (1989). From database to hypertext via electronic publishing: An information odyssey. In E. Barrett (Ed.), *The Society of Text: Hypertext, Hypermedia and the Social Construction of Information.* Cambridge, MA: MIT Press.

Brockman, H.J.D. (1993). Effective Communication with Interactive Media. *Information Services and Use,* (13) 183-187.

Brondmo, H. P. and Davenport, G. (1990). Creating and viewing the Elastic Charles: a hypermedia journal. In R. MacAleese, and C. Green (Eds), *Hypertext: State of the Art* (pp. 43-51). Oxford: Intellect.

Brooks, R.M. (1991). Comparative task analysis: An alternative direction for human-computer interaction science. In J. Carroll (Ed.), *Designing Interaction: Psychology at the Human-Computer Interface* (pp. 50-59). Cambridge: Cambridge University Press.

Brooks, R.M. (1993). Principles for effective hypermedia design. *Technical Communication, 40* (3 (August)) 423-429.

Brown, C. Marlin. (1988). *Human-Computer Interface Design Guidelines*. Norwood NJ: Ablex.

Brown, H. (1989a) (Ed.), *Proceedings, Hypermedia/Hypertext and Object-Oriented Databases* . Uxbridge: Unicom Seminars Ltd.

Brown, H. & Cole, F. (1989). The office document architecture & hypermedia. In H. Brown, (1989) (Ed.), *Proceedings, Hypermedia/Hypertext and Object-Oriented Databases* (pp. 10-19). Uxbridge: Unicom Seminars Ltd.

Brown, P.J. (1989b). Hypertext: Dreams and reality. In H. Brown (Ed.), *Proceedings, Hypermedia/Hypertext and Object-Oriented Databases* (pp. 60-71). Uxbridge: Unicom Seminars Ltd.

Brown, P.J. (1990). Assessing the quality of hypertext documents. In A. Rizk, N. Streitz, & J. Andre, (Eds), *Hypertext: Concepts, Systems & Applications* (pp. 1-12). Cambridge: Cambridge University Press.

Brown, P.J. (1992). Hypertext in Action: Some Results from an Application. *Information Services and Use,* 12, 41-54.

Bryan, M. (1994). Standards for text & hypermedia processing. *Information Systems & Use,* 13 (2) 93-102.

Buckingham , R.A., Hirschheim, R.A., Land, F.F. & Tully, C.J. (1987) (Eds) *Information Systems Education: Recommendations and Implementation* Cambridge: Cambridge University Press.

Buckingham, R.A., Hirschheim, R.A., Land, F.F. & Tully, C.J. (1987). Information systems curriculum: a basis for course design. In Buckingham et al (Eds) *Information Systems Education: Recommendations and Implementation*. Cambridge: Cambridge University Press.

Bulinger, H.J. (1991) (Eds), *Human Aspects in Computing: Design and Use of Interactive Systems and Information Management*. Elsevier Science Publishers.

Bush, V. (1945). As we may think. *Atlantic Monthly,* 176 (1) 101-108.

Buzan, T. (1974). *Use Both Sides of Your Brain*. NY: E.P. Dutton.

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

Card, S.K., Moran, T.P. & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Carlson, P. (1990). The rhetoric of hypertext. *Hypermedia,* 2(2) 109-131.

Carmel, E., Crawford, S.F., & Chen, H. (1992). Browsing in hypertext: A cognitive study. *IEEE Transactions on Systems, Man, and Cybernetics,* (22) 865-884.

Carroll, J. &. Tanner, P. (1987) (Eds), *Human Factors in Computing Systems - IV and Graphics Interface.* Amsterdam: ACM/North Holland.

Carroll, J. (1990). Infinite detail and emulation in an ontologically minimized HCI. In *Proceedings of the ACM CHI'90 conference (1990): Human Factors in Computing Systems.* Seattle, WA, 1-5 April (pp. 321-327).

Carroll, J. (1990). *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill.* Cambridge MA: MIT Press.

Carroll, J. M. (1991) (Ed.), *Designing Interaction: Psychology at the Human-Computer Interface.* Cambridge: Cambridge University Press.

Carroll, J. M., Smith-Kerker P.L., Ford J. R. & Mazur-Rimetz, S.A. (1988). The minimal manual. *Human-Computer Interaction,* (3) 123-153.

Carroll, J.M., Kellogg, W.A. & Rosson, M.B. (1991) The task-artefact cycle. In J. Carroll (Ed.). *Designing Interaction: Psychology at the Human-Computer Interface* (pp. 74-102). Cambridge: Cambridge University Press.

Cates, W.M. (1992). Fifteen Principles for Designing More Effective Instructional Hypermedia/Multimedia Products. *Educational Technology,* (32) 5-11.

Chalmers, A. (1976). *What is this thing called science?* Milton Keynes: Open University Press.

Checkland, P. & Scholes, J. (1990). *Soft Systems Methodology In Action.* Chichester: John Wiley & Sons.

Checkland, P. (1991). From Framework through experience to Learning: the essential nature of Action Research. In H.E. Nissen, H.K. Klein & R. Hirschheim (Eds).

Information Systems Research: Contemporary Approaches and Emergent Traditions (pp. 397-403). Amsterdam: Elsevier

Checkland, P. B. (1981). *Systems Thinking, Systems Practice*. Chichester, John Wiley.

Cherry, J.M., Turner, J.M., & Rockwell, G.M. (1992). Designing for diversity: The user interface for a hypermedia information system on a university campus. *Behaviour and Information Technology*, (11) 1-12.

Churcher, P.R. (1989). A common notation for knowledge representation, cognitive models, learning and hypertext. *Hypermedia*, 1(3) 235-254.

Conklin, J. (1987). Hypertext: an introduction & survey. *IEEE Computer*, 20(9) 17-41.

Cornford, T. & Smithson, S. (1996). *Project Research in Information Systems: A Student's Guide*. London: MacMillan Press.

Cotterman, W.W. & Senn, J.A. (1991) (Eds). *Challenges & Strategies For Research In Information Systems Development*. Chichester: John Wiley.

Coyne, R. (1995). *Designing technology in the post-modern age*. Cambridge, Massachusetts: MIT Press.

Cutts, G. (1991). *Structured systems analysis and design methodology* (2nd ed.). Oxford: Blackwell Scientific.

Dahlbom, B. & Mathiassen, L. (1993). *Computers in Context: The philosophy and practice of systems design*. Oxford: NCC Blackwell.

Darke, J. (1979). The primary generator and the design process. *Design Studies* (3) 157-162.

Davenport, E. (1994). Accounting for networked services: a preliminary report of fieldwork on UK computers. *Online Information '94 Proceedings*, (pp. 393-403).

Davidson, W.J. (1993). SGML authoring tools for technical communication. *Technical communication* . (Third quarter), 403-409.

Davis, G.B. (1992). Systems analysis and design: a research strategy macro-analysis. In W. W. Cottermann, & J.A. Senn, (Eds), *Challenges and strategies for research in systems development* (pp. 9-21). Chichester: John Wiley.

Dean, J.P. and Whyte, W.F. (1958). 'How do you know if the informant is telling the truth?' *Human Organisation*, (17) 34-38.

DeMarco, T. (1979). *Structured Analysis and System Specification*. Englewood Cliffs, NJ: Yourdon Press.

Denzin, N.K. & Lincoln, Y.S. (1994) (Eds), *Handbook of Qualitative Research*. Thousand Oaks: Sage.

DeRose, S.J. (1989). Expanding the notion of links. In *Proceedings ACM Hypertext '89 Conference*, Pittsburgh, PA, 5-8 November ACM (pp. 249-257).

Diaper, D. (1989) (Ed.), *Task Analysis for Human-Computer Interaction*. Chichester: Ellis Horwood.

Diaper, D. (1989). Task observation for human-computer interaction. In D. Diaper (Ed.), *Task Analysis for Human-Computer Interaction* (pp. 210-237). Chichester: Ellis Horwood.

Diaper, D. (ed.). (1989). *Knowledge Elicitation: Principles, Techniques and Applications*. Chichester: Ellis Horwood.

Diaper, D. and Winder, R. (Eds), *People and Computers III*. Cambridge: Cambridge University Press.

Diaper, D., Gilmore, D., Cockton, G. & Shackel, B. (1990) (eds), *Human-Computer Interaction: Interact '90*. London: McGraw-Hill.

Diaz, A. & Isakowitz, T. (1995). RMCase: Computer-aided support for hypermedia design and development. In F. Garzotto (Ed.), *Proceedings of the International Workshop on Hypermedia Design* . Montpellier, France, June 1995.

Dillon, A. & McKnight, C. (1995) Never mind the theory, feel the data: observations on the methodological problems of user interface design. In Schuler, W., Hannemann, J. & Streitz, N. (Eds) *ESPRIT Research Report Project 6532 HIFI* , Vol. 1. Springer-Verlag: Berlin.

Dillon, A. (1989). Human factors issues in the design of hypermedia interfaces. In H. Brown, (1989) (Ed.), *Proceedings, Hypermedia/Hypertext and Object-Oriented Databases* (pp. 43-50). Uxbridge: Unicom Seminars Ltd.

Dillon, A. (1990). Designing the human-computer interface to hypermedia applications. In D.H. Jonassen, & H. Mandl, (Eds), *Designing Hypermedia For Learning* (pp. 185-195). Berlin: Springer-Verlag.

Dillon, A. (1990). Review of Rivlin, C., Lewis, R & Davies Cooper, R. (1990) (Eds), *Guidelines for screen design*. In *Hypermedia*, 2 (2) 171-173.

Dillon, A. (1991). Requirements for hypertext applications: the why, what and how approach. *Applied Ergonomics,* 22(4 (Aug.)) 258-262.

Dillon, A. (1994). *Designing Usable Electronic Text: Ergonomic Aspects of Human Information Usage.* London: Taylor & Francis Ltd.

Dillon, A., McKnight, C. & Richardson, J. (1988). Reading from paper versus reading form screen. *The Computer Journal,* 31(5) 456 - 464.

Dimitroff, A. & Wolfram, D. (1993). Design Issues in a Hypertext-Based Information System for Bibliographic Retrieval. *Proceedings of the Asis Annual Meeting,* 30, (pp. 191-198).

Downton, A. (1991). Applied cognitive psychology: research for human-computer interaction. In A. Downton (Ed.). *Engineering the human-computer interface* (pp. 28-61). London: McGraw-Hill.

Downton, A. (1991). Engineering the human-computer interface: an overview. In A. Downton (Ed.). *Engineering the human-computer interface* (pp. 13-27). London: McGraw-Hill.

Draper, S. (1993). The notion of task. In S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel, and T. White, (Eds), *Bridges Between Worlds, INTERCHI'93 Conference Proceedings, Adjunct Proceedings* (pp. 207-208). Reading, MA: Addison-Wesley.

Duchastel, P. (1980). Research on illustrations in text: issues and perspectives. *Educational Communication and Technology,* 28(4) 283-287.

Duchastel, P.C. (1990). Examining cognitive processing in hypermedia usage. *Hypermedia,* 2(3) 221-233.

Duffy, T.M. & Jonassen, D.H. (1992). *Constructivism and the Technology of Instruction: A Conversation.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Duffy, T.M., Palmer, J.E. & Mehlenacher, B. (1992). *On-line Help: Design and Evaluation.* Norwood, N.J: Ablex.

Easterbrook, S. (1991). *What is Hypertext?* Brighton: School of Cognitive & Computing Sciences, Univ. of Sussex.

ECHT '92: Proceedings of the Fourth ACM Conference on Hypertext (1992)., INRIA, France, November 1992. ACM Press.

Egan, D., Remde, J., Gomez, L., Landauer, T., Eberhardt, J. & Lochbaum, C. (1989). Formative design-evaluation of SuperBook. *ACM Transactions on Information Systems,* 7(1 Jan.), 30-57.

Ellis, D., Ford, N. & Wood, F. (1993). Hypertext and learning styles. *The Electronic Library,* 11(1), 13-18.

Elm, W. &. Woods, D. (1985). Getting lost: a case study in interface design. In *Proceedings of the Human Factors Society 29th Annual Meeting* (pp. 927-31). Santa Monica, CA: Human Factors Society.

Evenson, S. & Rheinfrank, J. (1989). Towards a design language for representing hypermedia cues. In *Proceedings ACM Hypertext '89 Conference,* 5-8 November (pp. 83-92). Pittsburgh, PA: ACM

Farenc, C., Palenque, P. & Vanderdonckt, J (1995) User interface evaluation: is it ever usable? In Y. Anzai, K. Ogawam, and H. Mori, *Symbiosis of human and artefact,* Amsterdam: Elsevier Science B.V.

Forrester, M. D. (1990). HCI "intraface model" for system design. *Interacting with Computers,* 2(3) 279-296.

Foss, C. (1989). Tools for reading and browsing hypertext. *Information Processing and Management,* 25(4) 407-418.

Frisse, M. F. &. Cousins, S. B. (1992). Models of hypertext. *Journal of the American Society for Information Science,* 43(2) 183-191.

Furnas, G.W. (1986). Generalized fisheye views. In *Proceedings of ACM CHI'86 Conference.* Boston, MA, 13-17 April, (pp16-23).

Furuta, R. & Stotts, P.D. (1990). A functional meta-structure for hypertext models & systems. *Electronic Publishing,* 3(4),

Furuta, R. &. Stotts, P.D. (1990). Separating hypertext content from structure in Trellis. In McAleese, R and Green, C. (Eds), *Hypertext: State of the Art* (pp. 205-213). Oxford: Intellect.

Gaines, B. R. &. Vickers, J. M. (1988). Design Considerations for Hypertext Systems. *Microcomputers for Information Management,* 5(1 (March)), 1-27.

Galliers, R. (1992) (Ed.). *Information Systems Research: Issues, Methods and Practical Guidelines.* Oxford: Blackwell Scientific Publications.

Galliers, R.D. (1985) In search of a paradigm for information systems research In E. Mumford, R. Hirschheim, G. Fitzgerald & T. Wood-Harper (Eds). *Research Methods in Information Systems* (pp.281-297). Amsterdam: Elsevier Scientific Publications.

Gane, C. & Sarson, T. (1979). *Structured Systems Analysis: Tools & Techniques*. New York: Prentice-Hall.

Gardiner, W.L. (1991). From the Electronic Book through the Hyperbook to the BOOK. *Educational and Training Technology International*, (28) 302-305.

Gardner, L. & Paul, R. (1993). Developing a hypertext geographical information system for the Norfolk and Suffolk Broads Authority. *Hypermedia*, 5(2) 119-143.

Garrett, L., Smith, K. & Meyrovitz, N. (1987). Intermedia: issues, strategies and tactics in the design of a hypermedia document system. In *Proceedings of the Conference on Computer-Supported Co-operative Work* (pp. 163-174), Austin, TX, 3-5 December.

Garzotto F. (Ed.). (1995). *Proceedings of the International Workshop on Hypermedia Design*. Montpellier, France, June 1995.

Garzotto, F., Mainetti, L. & Paolini, P. (1995). Hypermedia design environments and the hypertext design. *Communications of the ACM*, August, 38(8) 74-87.

Garzotto, F., Paolini, P., & Schwabe, D. (1993). HDM - A model-based approach to hypertext application design. *ACM Transactions on Information Systems, 11*(1), 1-26.

Garzotto, F., Paolini, P., Schwabe, D. & Bernstein, M. (1991). Tools for Designing hyperdocuments. In E. Berk, & J. Devlin, (Eds). *The Hypertext/Hypermedia Handbook* (pp. 179-207). NY: Intertext Publications/ McGraw Hill.

Gay, G., Trumbull, D., & Mazur, J. (1991). Designing and testing navigational strategies and guidance tools for a hypermedia program. *Journal of Educational Computing Research*, 7, 189-202.

Gentner, D. (1990). Why good engineers sometimes create bad interfaces. In *Proceedings of the ACM CHI'90 Conference* (pp. 277-282). Human Factors in Computing Systems, Seattle, WA, 1-5 April.

Glushko, R. & Kershner, K. (1993). Silicon Graphics' IRIS Insight: an SGML success story. *Technical Communication*, (Third quarter) 394-402.

Glushko, R. (1990). Designing a hypertext electronic encyclopedia. *Bulletin of the American Society for Information Science, 16*(3 (February-March), 14-22.

Glushko, R. (1992). Seven Ways to Make a Hypertext Project Fail. *Technical Communication, 39*(2), 226-230.

Glushko, R. (1992). Using Off-the-shelf Software to Create a Hypertext Electronic Encyclopedia. *Technical Communications,* 39,(2), 226-230.

Goodman, D. (1994). *The Complete HyperCard 2.0 handbook. (3rd ed.).* NY: Bantam.

Gould, J. D. (1988). How to design usable systems. In M. Helander (Eds), *Handbook of Human-Computer Interaction* (pp. 757-789): Amsterdam: Elsevier Science (North-Holland).

Gould, J.D., Alfaro, L., Finn, R., Haupt, B. & Minuto, A. (1987) Reading from CRT displays can be as fast as reading from paper. *Human Factors* 29(5), 497-517.

Greenbaum, J. & Kyng, M. (1991) (Eds). *Design at Work: Cooperative Design of Computer Systems* Hillsdale, N J: Laurence Erlbaum Associates.

Grice, R.A. & Ridgway, L.S. (1993). Usability and Hypermedia: Toward a Set of Usability Criteria and Measures. *Technical Communication,* (40), 429-437.

Grief, S. (1991). The role of German work psychology in the design of artefacts. In J. Carroll (Ed.). *Designing Interaction: Psychology at the Human-Computer Interface* (pp. 203-226). Cambridge: Cambridge University Press.

Grøenbaek, K. &. Trigg, R. (1994). Design issues for a Dexter-based hypermedia system. *Communications of the ACM, 37*((2) February), 40-49.

Grudin, J., Ehrlich, S. & Shriner, R. (1987). Positioning human factors in the user interface development chain. In J. Carroll & P. Tanner (Eds). *Human Factors in Computing Systems - IV and Graphics Interface* (pp. 125-131). Amsterdam: ACM/North-Holland.

Guindon, R. & Curtis, B. (1988) Control of cognitive processes during design: what tools would support software designers? In *Proceedings of CHI '88* Washington, DC: ACM Press, Addison -Wesley Publishing.

Guindon, R. (1989). The process of knowledge discovery in system design. In G. Salvendy & M. Smith (Eds). *Designing and Using Human-Computer Interfaces and Knowledge-Based Systems* (pp. 727-734). Amsterdam: Elsevier.

Guindon, R., Krasner, H. & Curtis, B. (1987). Breakdowns and processes during the early activities of software design by professionals. In G. Olson, E. Soloway & S. Sheppard (Eds), *Empirical Studies of Programmers, Second Workshop.* Ablex Publishing Corp.

Gummesson, E. (1988) *Qualitative Methods in Management Research.* Bromley: Chartwell-Bratt.

Halasz, F. & Schwartz, M. (1990). The Dexter hypertext reference model. In *Proceedings NIST Hypertext Standardization Workshop*,(pp 95-133). Gaithersberg MD 16-18 January, .

Halasz, F. & Schwartz, M. (1994). The Dexter hypertext reference model. *Communications of the ACM, 1994*, (Feb.), 30-39.

Halasz, F. (1988). Reflections on NoteCards: seven issues for the next generation of hypermedia systems. *Communications of the ACM, 31*(7), 836-852.

Halasz, F., Moran, T. & Trigg, R. (1987). NoteCards in a nutshell. In *ACM Conference on Human Factors in Computing Systems: CHI+GI 87* (pp. 45-52). Toronto, Canada, 15-19 May.

Hamilton, S. (1992). MIS research strategies. In R. Galliers (Ed.), *Information Systems Research - Issues, Methods and Practical Guidelines* (pp. 132-143). Oxford: Blackwell.

Hammersley, P. (1991). Information systems design methodologies - book reviews. *The Computer Journal*, 34 (2), 182-185.

Hammond, N. (1996). Auto-moitoring: theoretical touchstone or circular catch-all? Http://www.icbl.hw.ac.uk/~granum/class/alt6docs/nic_alt.htm.

Hammond, N. & Allinson, L. (1988). Development and evaluation of a CAL system for non-formal domains: the hitch-hiker's guide to cognition. *Computers in Education*, 12 1, 215-220.

Hardman, L. & Sharratt, B. (1990). User-centred hypertext design: the application of HCI design principles and guidelines. In R. McAleese, &. C. Green (Eds), *Hypertext: State of the Art* (pp. 252-259). Oxford: Intellect.

Hardman, L. (1989). Evaluating the usability of the Glasgow Online hypertext. *Hypermedia*, 1(1), 34-63.

Hardman, L., Bulterman, D., & Van Rossum, G. (1994). The Amsterdam Hypermedia model. *Communications of the ACM* 37(2), 50-63.

Harris, R. A. & Hosier, W. J. (1991). A taxonomy of online information. *Technical Communication, 1991* (Second quarter), 197-210.

Helander, M. (1988) (Ed.), *Handbook of Human-Computer Interaction* North-Holland: Elsevier.

Heller, R. (1991). Computing access in public spaces: design lessons learned. In H.J. Bulinger (Ed.), *Human Aspects in Computing: Design & Use of Interactive Systems and Information Management* (pp. 699-703). Amsterdam: Elsevier.

Henderson, A. (1991). A development perspective on interface, design, and theory. In J. Carroll (Ed.), *Designing Interaction: Psychology at the Human-Computer Interface* (pp. 254-268). Cambridge: Cambridge University Press.

Herrstrom, D. S. and Massey, D. G. (1989). Hypertext in context. In E. Barrett (Ed.), *The Society of text: Hypertext, Hypermedia and the Social Construction of Information* (pp. 45-58). Cambridge, MA: MIT Press.

Hewett, T. T. (1987). The Drexel Disk: an electronic 'guidebook'. In D. Diaper , and R. Winder (Eds), *People and Computers III* (pp. 115-129). Cambridge: Cambridge University Press.

Hewett, T. T. (1989). The Drexel Disk: Hypertext-based instructional software as a tool for the exploration of a constrained knowledge space. In *Behavior Research Methods, Instruments, and Computers. 18th Annual Meeting of the Society for Computers in Psychology (1988, Chicago, Illinois).* (21), 316-325.

Hill, G., Wilkins, R., & Hall, W. (1993). Open and reconfigurable hypermedia systems: a filter-based model. *Hypermedia,* 5(2), 103-118.

Hirschheim, R. & Klein, H. (1992). Paradigmatic influences on information systems development methodologies: evolution and conceptual advances. *Advances in Computers,* (34), 294-392.

Hirschheim, R. (1985). Information systems epistemology: an historical perspective. In E. Mumford, R. Hirschheim, G. Fitzgerald & T. Wood-Harper (Eds), *Research Methods in Information Systems* (pp 13-35). Amsterdam: Elsevier Scientific Publications.

Hix, D. & Hartson, H.R. (1993). *Developing User Interfaces: Ensuring Usability Through Product and Process.* New York: John Wiley.

Hodges, M. E. &. Sasnett, R. M. (1993). *Multimedia Computing: Case Studies from MIT Project Athena.* Reading MA: Addison-Wesley.

Hodges, M., Sasnett, R. & Ackerman, M. (1989). A construction set for multimedia applications. *IEEE Software,* 6 (1), 37-43.

Holt, P. O. & Williams, N. (1992) (Eds) *Computers and Writing: State of the Art.* Oxford: Intellect.

Horn, R.E. (1989). *Mapping Hypertext: Analysis, Linkage and Display of Knowledge for the Next Generation of On-Line Text and Graphics*. Lexington, MA: The Lexington Institute.

Horton, W. (1990). *Designing and Writing On-line Documentation* (1st ed.). New York: John Wiley.

Horton, W. (1994). *Designing and Writing On-line Documentation* (2nd ed.). New York: John Wiley.

Howard, T. (1993). Electronic Distribution of hypermedia on wide-area networking systems: an update. *Technical Communication* (Third quarter), 438-448.

Hutchings, G.A., Carr, L.A. & Hall, W. (1992). Stackmaker: An environment for creating hypermedia learning material. *Hypermedia*, 4(3), 197-211.

Hutchings, G.A., Hall, W., Briggs, J., Hammond, N.V., Kibby, M.R., McKnight, C. & Riley, D. (1992). Authoring and evaluation of hypermedia for education. *Computers and Education*, 18 1(3), 171-177.

Iivari, J. (1991) A paradigmatic analysis of contemporary schools in IS development. *European Journal of Information Systems*. 1 (4), 249-272.

Isakowitz, T., Stohr, E. A. & Balusubramanian, P. (1995). RMM: a methodology for structured hypertext design. *Communications of the ACM*, 38(8 (Aug.)), 34-43.

Jackson, M. (1983). *Systems Development*. Englewood Cliffs NJ: Prentice-Hall.

Jayaratna, N. (1994). *Understanding and Evaluating Methodologies: NIMSAD: A Systematic Framework* (1 ed.). London: McGraw-Hill.

Jaynes, J. T. (1989). Limited freedom: linear reflections on non-linear texts. In E. Barrett (Ed), *The Society of Text: Hypertext, Hypermedia and the Social Construction of Information* Cambridge, MA: MIT Press.

Johanson, W.W.F. (1993). The electronic advantage. *Case Currents, 1993*(October), 46-49.

Johnson, P. (1992). *Human-Computer Interaction: Psychology, Task Analysis & Software Engineering*. London: McGraw-Hill.

Jonassen, D. H. (1986). Hypertext principles for text and courseware design. *Educational Psychologist, Special Issue: Design of Instructional Text*. 21(4), 269-292.

Jonassen, D. H. (1988). Designing structured hypertext and structuring access to hypertext. *Educational Technology, 1988*(November), 13-16.

Jonassen, D. H. (1991). Hypertext as instructional design. *Educational Technology Research & Development*, (39), 83-92.

Jonassen, D.H. (1996). Concept mapping and other formalisms as mindtools for representing knowledge.
Http://www.icbl.hw.ac.uk/~granum/class/alt6docs/dav_alt.htm

Jonassen, D. H., Hannum, W. & Tessmer, M. (1989). *Handbook Of Task Analysis Procedures*. NY: Praeger.

Jonassen, D.H. & Mandl, H. (1990) (Eds). *Designing Hypermedia For Learning*. Berlin: Springer-Verlag.

Jones, J. C. (1970). *Design Methods: Seeds of Human Futures*. (1st ed.). New York: John Wiley.

Jones, J. C. (1980). *Design Methods: Seeds of Human Futures*. (2nd ed.). New York: John Wiley.

Jonsson, S. (1991). Action research. In H.E. Nissen, H.K. Klein & R. Hirschheim (Eds). *Information Systems Research: Contemporary Approaches & Emergent Traditions* (pp. 371-396). Amsterdam: Elsevier

Jordan, D., Russell, D., Jensen, A. & Rogers, R. (1989). Facilitating the development of representations in hypertext with IDE. In *Proceedings ACM Hypertext '89 Conference*, Pittsburgh, PA, 5-8 November (pp. 93-104). ACM.

Jorgensen, A. (1989). Using the thinking-aloud method in system development. In G. Salvendy & M. Smith (Eds). *Designing and Using Human-Computer Interfaces and Knowledge-Based Systems* (pp. 743-750). Amsterdam: Elsevier.

Kahn, P. (1989). Webs, trees, and stacks: How hypermedia system design effect hypermedia content. In G. Salvendy, & M. Smith, (Eds), *Designing and Using Human-Computer Interfaces and Knowledge-Based Systems* (pp. 443-449). Amsterdam: Elsevier.

Kahn, P. and Lenk, K. (1993). Typography for the computer screen: applying the lessons of print to electronic documents. *The Seybold Report on Desktop Publishing*, (11), 3-16.

Kahn, P., Peters, R. & Landow, G.P. (1995) Three fundamental elements of visual rhetoric in hypertext. In Schuler, W. Hannemann, J. & Streitz N. (Eds), *ESPRIT Research Report* Project 6532 HIFI Vol. 1 Berlin: Springer-Verlag.

Kearsley, G. (1988). Authoring considerations for hypertext. *Educational Technology, 1988* (November) 21-24.

Keen, C. and Higgins, T. (1990). *Young People's Knowledge Of Higher Education*. Leeds: Heist.

Keen, C. and Higgins, T. (1992) *Adults' Knowledge Of Higher Education*. Leeds: Heist.

Keys, P. (1988). A methodology for methodology choice. *System Research*, 5(1), 65-76.

Kinzie, M. & Berdel, R. (1990). Design and use of hypermedia systems. *Educational Technology, Research & Development*, 38(3), 61-68.

Kirwan, B. & Ainsworth, L.K. (1992) *A Guide to Task Analysis* London: Taylor & Francis.

Knauper, A. (1989). Rapid prototyping for design and evaluation of human-machine interfaces. In G. Salvendy & M.J. Smith (Eds). *Designing And Using Human-Computer Interfaces And Knowledge-Based Systems* (pp. 549-556). Amsterdam: Elsevier Science Publishers B.V.

Knee, M. & Atkinson, S.D. (1990). *Hypertext/Hypermedia: an Annotated Bibliography*. Westport: Greenwood Press.

Knussen, C., Tanner, G., & Kibby, M. (1991). An approach to the evaluation of hypermedia. *Computers and Education*, 17(1), 13-24.

Knuth, R.A. and Brush, T. A (1990). Results of the Hypertext '89 design survey. *Hypermedia*, 2(2), 91-107.

Kommers, P.A.M. (1991). Virtual structures in hypermedia resources. In H.J. Bulinger (Ed.), *Human Aspects in Computing: Design and Use of Interactive Systems and Information Management* (pp. 1343-1351). Amsterdam: Elsevier.

Kotler, P. (1988) *Marketing Management; Analysis, Planning , Implementation and Control* (6th ed.). NJ: Prentice Hall.

Kreitzberg, C. (1989). Designing the electronic book. In G. Salvendy & M.J. Smith (Eds), *Designing And Using Human-Computer Interface And Knowledge-Based Systems* (pp. 457-464). Amsterdam: Elsevier.

Kuhlthau, C.C. (1988). Longitudinal Case Studies of the Information Search Process of Users in Libraries. *Library and Information Science Research*, (10), 257-304.

Landauer, T. (1987). Psychology as a mother of invention. In J. Carroll & P. Tanner (Eds), *Human Factors in Computing Systems - IV and Graphics Interface* (pp. 333-335). Amsterdam: ACM/North-Holland.

Landauer, T. (1991). Let's get real: a position paper on the role of cognitive psychology in the design of humanly useful and usable systems. In J. M. Carroll, (Ed.), *Designing Interaction: Psychology at the Human-Computer Interface.* Cambridge: Cambridge University Press.

Landow, G.P. (1989). The rhetoric of hypertext: some rules for authors. *Journal of Computing in Higher Education,* 1(1), 39-63.

Langford, D. &. Brown, P. (1993). Creating hypertext documents: is it worth it? *Aslib Proceedings,* 45(4), 91-95.

Langford, D. (1993). Evaluating a hypertext document. *Aslib Proceedings,* 45(9), 221-226.

Latchem, C., Williamson, J. & Henderson-Lancett, L. (1993) (Eds). *Interactive Multimedia - Practice and Promise.* London: Kogan-Page.

Latta, G.F. & Swigger, K. (1992). Validation of the Repertory Grid for Use in Modelling Knowledge. *Journal of the American Society for Information Science,* (43), 115-129.

Laurel, B. (1990). *The Art of Human-Computer Interface Design.* Reading MA: Addison-Wesley.

Laurel, B., Oren, T. & Don, A. (1990). Issues in multimedia interface design: media integration & interface agents. In *Proceedings of the ACM CHI'90 Conference* Human Factors in Computing Systems, Seattle, WA, 1-5 April. (pp. 133-139). ACM.

Lawson, B. (1980). *How Designers Think.* London: Architectural Press.

Leigh, W. & Paz, N. (1991). Using Visual Programming to Construct an Information Retrieval Tool for Student Advising. *Microcomputers for Information Management,* 8, 197-209.

Locke, L.F., Spirduso, W.W. & Silverman, S.J. (1992). Developing proposals: some common problems. In R. Galliers (Ed.). *Information Systems Research: Issues, Methods and Practical Guidelines* (pp. 182-251). Oxford: Blackwell Scientific Publications.

Longworth, G. & Nicholls, D. (1987). *The SSADM Manual.* Manchester: NCC.

Lucarella, D. & Zanzi, A. (1993). Information Retrieval from Hypertext: An Approach Using Plausible Inference. *Information Processing and Management,* 29, 299-312.

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

Marchionini, G. (1989). Making the transition from print to electronic encyclopedia: adaptation of mental models. *International Journal of Man-Machine Studies*, 30(6), 591-618.

Marchionini, G. (1992). Interfaces for End-User Information Seeking. *Journal of the American Society for Information Science* 43(2), March), 156-163.

Marlow, E. (1993). Electronic communications media: massaging the messages. *Technical Communication* (Third quarter), 453-456.

Martin, J. (1990). *Hyperdocuments and How to Create Them* . Englewood Cliffs, NJ: Prentice Hall.

Mayer, R. E. (1988). From novice to expert. In M. Helander (Ed), *Handbook of Human-Computer Interaction*. Amsterdam: North-Holland.

Mayhew, D.J. (1992). *Principles and Guidelines in User Interface Design*. Englewood Cliffs NJ: Prentice-Hall.

McAleese, R. (1989) (Ed.), *Hypertext: Theory into Practice*. Oxford: Intellect,

McAleese, R. &. Green, C. (1990) (Eds), *Hypertext: State of the Art*, Oxford: Intellect.

McAleese, R. (1996). A theoretical view on concept mapping. Http://www.icbl.hw.ac.uk/~granum/class/alt6docs/ray_alt.htm.

McFarlan, F.W. (1984) (Ed.), *The Information Systems Research Challenge - Proceedings*. Boston: Harvard Business School Press.

McKerlie, D. &. Preece, J. (1993). The hype and the media: issues concerned with designing hypermedia. *Journal of Microcomputer Applications*, (16), 33-47.

McKnight, C., Dillon, A. & Richardson, J. (1989) Problems in hyperland? A human factors perspective. *Hypermedia*, 1(2), 167-178.

McKnight, C., Dillon, A. & Richardson, J. (1990). A comparison of linear and hypertext formats in information retrieval. In R. McAleese &. C. Green (Eds), *Hypertext: State of the Art* (pp. 10-19). Oxford: Intellect.

McKnight, C., Dillon, A. & Richardson, J. (1991). *Hypertext in Context*. Cambridge: Cambridge University Press.

McKnight, C., Richardson, J. & Dillon, A. (1989). The authoring of hypertext documents. In R. McAleese, (1989) (Ed.), *Hypertext: Theory into Practice*. Intellect, Oxford. (pp. 139-147).

Megarry, J. (1991). "Europe in the Round": Principles and Practice of Screen Design. *Educational and Training Technology International,* (28), 306-315.

Meyrovitz, N. (1986). Intermedia: the architecture and construction of an object-oriented hypermedia system and applications framework. In *OOPSLA '86 Proceedings* (pp. 186-201). NY: ACM.

Meyrovitz, N. (1989). The missing link: why we're all doing hypertext wrong. In E. Barrett (Ed.), *The Society of Text: Hypertext, Multimedia and the Social Construction of Information.* Cambridge, MA: MIT Press.

Michon, B. (1993). Hotspots. In M.E. Hodges, & R. M. Sasnett, (Eds), *Multimedia Computing: Case Studies from MIT Project Athena* (pp. 219-227). Reading MA: Addison-Wesley.

Milheim, W.D. & Lavix, C. (1992). Screen Design for Computer-Based Training and Interactive Video: Practical Suggestions and Overall Guidelines. *Performance and Instruction,* (31), 13-21.

Mingers, J. (1996). *Combining Research Methods in Information Systems: Multi-Paradigm Methodology* (Research report No. 239). Warwick Business School.

Mitchell, J. Clyde (1983). Case and situation analysis. *Sociological Review,* 31(2), 187-211.

Molich, R. & Nielsen, J. (1990). Improving a human-computer dialogue: what designers know about traditional interface design. *Communications of the ACM,* 33(3), 338-348.

Moline, J. (1991). Designing Multimedia Systems for Museum Objects and Their Documentation. *Microcomputers for Information Management,* (8), 69-86.

Morgan, P. (1991). Hypertext and the literary document. *Journal of Documentation,* (47), 373-388.

Mullins, P.M. & Treu, S. (1993) A task-based cognitive model for user-network interaction: defining a task taxonomy to guide the interface designer. *Interacting with computers.* 5(2), 139-166.

Mumford E. (1983) *Designing Participatively.* Manchester: Manchester Business School.

Mumford, E. & Henshall, D. (1979) *A Participative Approach to Computer Systems Design,* London: Associated Business Press.

Mumford, E., Hirschheim, R., Fitzgerald, G. & Wood-Harper, T. (Eds) (1985). *Research Methods in Information Systems.* Amsterdam: Elsevier Scientific Publications.

Nelson, M.J. (1991). The Design of a Hypertext Interface for Information-Retrieval. *Canadian Journal Of Information Science*, 16(2), 1-12.

Nicholls, D. (1987). *Introducing SSADM: the NCC Guide*. Manchester: NCC Publications.

Nielsen, J (1995). *Multimedia and Hypertext: The Internet and Beyond*. Cambridge, MA: Academic Press.

Nielsen, J. & Molich, R. (1990). Heuristic evaluation of user interfaces. In *Proceedings of the ACM CHI'90 Conference* Human Factors in Computing Systems, Seattle, WA, 1-5 April (pp. 249-255). ACM.

Nielsen, J. (1989). The matters that really matter for hypertext usability. In *Proceedings ACM Hypertext '89 Conference*, Pittsburgh, PA, 5-8 November (pp. 239-249). ACM.

Nielsen, J. (1989). Usability engineering at a discount. In G. S. Salvendy & M.J. Smith (Eds), *Designing and Using Human-Computer Interfaces and Knowledge-Based Systems* (pp. 394-401). Amsterdam: Elsevier Science.

Nielsen, J. (1990). Evaluating hypertext usability. In D.H. Jonassen & H. Mandl. *Designing Hypermedia for Learning* (pp. 147-168). NATO ASI Series Vol. F67. Berlin: Springer-Verlag.

Nielsen, J. (1990). *Hypertext and Hypermedia*. London: Academic Press.

Nielsen, J., & Lyngbaek, U. (1990). Two field studies of hypermedia usability. In R. McAleese, & C. Green, (Eds), *Hypertext: State of the Art* (pp. 64-72).

Nissen, H.E., Klein, H.K. & Hirschheim, R. (Eds). (1991). *Information Systems Research: Contemporary Approaches and Emergent Traditions*. Amsterdam: Elsevier (North-Holland).

Nix, D & Rand, J.S. (Eds). (1990). *Cognition, Education and Multimedia: Exploring Ideas in High Technology*. Lawrence Erlbaum Associates: Hillsdale, NJ.

Norman, D. (1991). Cognitive artefacts. In J. Carroll (Ed.), *Designing Interaction: Psychology at the Human-Computer Interface*. Cambridge: Cambridge University Press

Norman, D. A. &. Draper, S.W. (1986). *User-Centered System Design*. Hillsdale, NJ: Lawrence Erlbaum, Associates

Nowaczk, R. &. Snyder, A. (1993). Using hypertext for college course information. *Computers in Education*, 21(4), 273-279.

Olle, T.W., Hagelstein, J., McDonald, I.G., Rollands, C., Sol, H.G., Van Assche, F.J.M. & Verrijn-Stuart, A.A. (1993). *Information Systems Methodologies: A Framework for Understanding*. Wokingham: Addison-Wesley.

Olle, T.W., Sol, H.G. & Tully, C.J. (1983). *Information Systems Design Methodologies: A Feature Analysis*. Amsterdam: North-Holland.

Ong, W.J. (1982). *Orality and Literature: The Technologising of the Word*. London: Routledge.

Oren, T. (1987). The architecture of static hypertexts. In *Proceedings of ACM Hypertext '87 Conference* 13-15 November (pp291-306). Chapel Hill, NC.

Paranuk, H.V. (1994). Ordering the information graph. In E. Berk & J. Devlin (Eds), *The Hypertext/Hypermedia Handbook* (pp. 299-325). New York: Intertext Publications/McGraw Hill.

Park, I. & Hannafin, M.J. (1993). Empirically-Based Guidelines for the Design of Interactive Multimedia. *Educational Technology, Research and Development*, (41), 63-85.

Parsaye, K., Chignell, M., Koshafian, S. & Wong, H. (1989). *Intelligent Databases: Object-Oriented Deductive Hypermedia Technologies*. New York: John Wiley.

Payne, S. & Green, T.R.G. (1989) Task-action grammar: the model and its development. In D. Diaper, (Ed.), *Task Analysis for Human-Computer Interaction*. Chichester: Ellis Horwood.

Perez, E. (1991). Tools for authoring hypertexts. In E. Berk, & J. Devlin, (Eds), *The Hypertext/Hypermedia Handbook*, (pp. 209-224). NY: Intertext Publications/McGraw Hill.

Perlman, G. (1989). Asynchronous design/evaluation methods for hypertext technology development. In *Proceedings ACM Hypertext '89 Conference*, (pp. 61-81) Pittsburgh, PA, 5-8 November. ACM.

Perlman, G. (1990). Evaluating hypermedia systems. In *Proceedings of the ACM CHI'90 Conference Human Factors in Computing Systems*, 1-5 April pp. (387-390). Seattle, WA.

Platt, J. (1988). What case studies do. In *Studies in Qualitative Methodology* (1) pp. 1-23. JAI Press Inc.

Popper, K. (1986) *The Poverty of Historicism*. Reading: ARK. (Originally published in 1957).

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

Proceedings of ACM Hypertext '87 Conference (1987).Chapel Hill, NC., 13-15 November. ACM.

Proceedings of ACM Hypertext '89 Conference, (1989), November. Pittsburgh, PA, 5-8 . ACM.

Proceedings of ACM Multimedia '94 Towards Usability Guidelines for Multimedia Systems, (Oct. 15-20 1994) San Francisco, CA.

Proceedings of CHI & GI '87: Conference(1987) Human factors and graphical interface. New York: ACM.

Proceedings of CHI '88 Washington, DC: ACM Press, Addison-Wesley Publishing.

Proceedings of the ACM CHI'90 Conference (1990) Human Factors in Computing Systems 1-5 April. Seattle, WA.

Proceedings of the Conference on Computer-Supported Co-operative Work Austin, TX, 3-5 December.

Proceedings of the Workshop on Hypermedia Design and Development at the European Conference on Hypermedia Technologies (1994). (ECHT '94). Technical report IS-94-8. New York: New York University.

Proceedings, ISTIP-91 - Fourth UK Conference on Information Systems Teaching: Improving the Practice - Sunningdale 3-5 April 1991.

Pulst, E., Wolfram, G., Leclerc, A., Puerta, M.C., Manfredi, F., Bleecke, J., Kieback, A., Gorner, C., Klein, B., Nemeier, J., Belsmuller, E., Fresko, M., Conrath, D., Hawgood, J., Leppanen, M. & Savolainen, V. (1990). *Hectors's Framework of Reference. Report of ESPRIT Project 2082*. Stuttgart: HECTOR, Fraunhofer Institute IAO.

Rada, R. & You, G.N. (1992). Balanced Outlines and Hypertext. *Journal of Documentation*, 48, 20-44.

Rada, R. (1989). Writing and reading hypertext: An overview. *Journal of the American Society for Information Science*, (May), 164-171.

Rada, R. (1991). Small, medium & large hypertext. *Information Processing & Management*, 27(6), 659-677.

Rada, R. (1992). Converting a textbook to hypertext. *ACM Transactions on Information Systems*, 10(3), 294-315.

Ramaiah, C. (1993). Use of hypertext for teaching and training: a bibliography. *The Electronic Library*, 11(6), 409-417.

Rao, U. & Turoff, M. (1990). Hypertext functionality: a theoretical framework. *International Journal of Human-Computer Interaction*, 2(4), 333-357.

Raskin, J. (1987). The hype in hypertext: a critique. In *Proceedings of ACM Hypertext '87 Conference (1987)* Chapel Hill, NC, 13-15 November. (325-330).

Reeves, T. (1993). Research support for interactive multimedia: existing foundations and new directions. In C. Latchem, J. Williamson & L. Henderson-Lancett (Eds). *Interactive Multimedia - Practice and Promise* (pp. 79-96). London: Kogan-Page.

Richards, T., Chignell, M., & Lacey, R. (1990). Integrating hypermedia: bridging the missing links. *Academic Computing*, (Jan), 24-44.

Rizk, A., Streitz, N. & Andre, J. (Eds). (1990). *Hypertext: Concepts, Systems & Applications*. Cambridge: Cambridge University Press.

Robinson, B. (1992). *Mixed Mode Document Research: The Collected Reports*. London: British Library R & D Department.

Rosson, M., Maass, S. & Kellogg, W. (1987). Designing for designers: An analysis of design practice in the real world. In J. M. Carroll. &. P. P. Tanner (Eds), *Human Factors in Computing Systems - IV and Graphics Interface: Proceedings of the CHI + GI '87 Conference, Toronto Canada 5-9 April 1987* (pp. 137-142). Amsterdam: ACM/North Holland.

Rouse, W. B. &. Rouse, S. H. (1984). Human information seeking and design of information systems. *Information Processing and Management*, 20(1-2), 129-138.

Rowley, J.E. (1993). Information systems methodologies: a review & assessment of their applicability to the selection, design and implementation of library & information systems. *Journal of Information Science*, 19(4), 291-302.

Rubottom, J. (1993). Using Lotus SmarText to build hypertext documents. *Computers in Libraries*, 13(3), 76

Runkel, P. (1990). *Casting Nets and Testing Specimens: Two Grand Methods of Psychology* New York: Praeger.

Rzevski, G. (1983). On the comparison of design methodologies. In T.W. Olle, H.G. Sol & C.J. Tully (Eds). *Information Systems Design Methodology: A Feature Analysis* (pp. 259-266).

Salminen, A. & Watters, C. (1992). A Two-Level Structure for Textual Databases to Support Hypertext Access. *Journal of the American Society for Information Science*, 43, 432-447.

Saloman, G. (1990). Designing casual-use hypertext: the CHI '89 InfoBooth. In *Proceedings of the ACM CHI'90 Conference* Human Factors in Computing Systems, Seattle, WA, 1-5 April (pp. 451-458). ACM.

Salvendy, G. & Smith, M. (Eds). (1989). *Designing and Using Human-Computer Interfaces and Knowledge-Based Systems*. Amsterdam: Elsevier.

Sauer, C. (1993). *Why Information Systems Fail: a Case Study Approach*. Henley-on-Thames: Alfred Waller.

Savolainen, V. (1991). Comparison analysis of design methodologies and reference frameworks for information systems design. In H. J. Bulinger (Eds), *Human Aspects in Computing: Design and Use of Interactive Systems and Information Management* (pp. 1135-1139). Amsterdam: Elsevier Science Publishers.

Scheller, A. (1993). The Open Document Architecture (ODA) and Its HyperODA Extensions. *Information Services and Use*, (13), 121-130.

Schuler, W., Hannemann, J. & Streitz, N. (Eds). *ESPRIT Research Report Project 6532 HIFI* Vol. 1. Springer-Verlag: Berlin.

Schwabe, D. & Rossi, G. (1995). The object-oriented hypermedia design model. *Communications of the ACM* 38 (8) 45-46.

Schwabe, D., Caloini, A., Garzotto, F., & Paolini, P. (1992). Hypertext development using a model-based approach. *Software - Practice and Experience*, 22(11 (Nov.)), 937-962.

Scottish Office Education Department. (1993) *The Further and Higher Education Charter: Report on Research Among Students, Parents and the General Public in Scotland*. London: The British Market Research Bureau, Survey Research Division.

Search, P. (1993). Hyperglyphs: using design & language to define hypermedia navigation. *Technical Communication* (Third quarter), 415-421.

Sharples, M. and Pemberton, L. (1990). *Starting from the Writer: Guidelines for the Design of User-Centred Document Processors* (Cognitive Science Research Paper No. CSRP 154). School of Cognitive and Computing Sciences, University of Sussex.

Sharples, M., Goodlet, J., Beck, E., Wood, C., Easterbrook, S., Plowman, L. & Evans, W. (1990). *A Framework for the Study of Computer-Supported Collaborative Writing*.. Working paper, School of Cognitive and Computing Sciences, University of Sussex.

Shneiderman, B & Kearsley, G. (1989) *Hypertext: Hands-On! An Introduction to a New Way of Organising and Accessing Information*. Reading MA: Addison-Wesley.

Shneiderman, B. (1987). *User Interface Design and Evaluation for an Electronic Encyclopedia*. Technical report CS-TR-1819 Dept. of Computer Science, University of Maryland.

Shneiderman, B. (1988). Finding facts versus browsing knowledge in hypertext systems. *IEEE Computer*, Jan, 70-80.

Shneiderman, B. (1989). Reflections on authoring, editing and managing hypertext. In E. Barrett (Ed.), *The Society of Text: Hypertext, Hypermedia and the Social Construction of Information* (pp. 115-131). Cambridge MA: MIT.

Shneiderman, B. (1992). *Designing the User Interface* (2nd ed.). Reading, MA: Addison-Wesley.

Shneiderman, B., Kreitzberg, C. & Berk, E. (1991). Editing to structure a reader's experience. In E. Berk, & J. Devlin, (Eds). *The Hypertext/Hypermedia Handbook*. NY: Intertext Publications/McGraw Hill.

Simon, H. A. (1981). *The Sciences of the Artificial*. Cambridge MA: MIT Press.

Simpson, A. & McKnight, C. (1990). Navigation in hypertext: structural cues and mental maps In R. McAleese & C. Green (Eds), *Hypertext: State of the Art* (pp. 73-83). Oxford: Intellect.

Smith, P. & Wilson, J. (1993). Navigation in hypertext through virtual environments. *Applied Ergonomics*, 24(4), 271-278.

Smith, S. & Mosier, J. (1986). *Guidelines for Designing User Interface Software* - Report ESD-TR-86-278, Electronic software division, the Mitre Corporation. Bedford MA/Springfield VA: The Mitre Corporation/NTIS.

Smith, S. L. (1988). Standards versus guidelines for designing user interface software. In M. Helander (Ed.), *Handbook of Human-Computer Interaction* (pp. 877-889). North-Holland: Elsevier.

Sobiesak, R. & Mylopoulos, J. (1991). A conceptual modelling approach to authoring-in-the-large for hypertext documents. *SIGOIS Bulletin*, 12 (2-3), 225-239.

Sommerville, I. (1992). *Software Engineering*. (4th ed.). Wokingham: Addison-Wesley.

Spiro, R.J. & Jehng, J.C. (1990). Cognitive flexibility and hypertext: Theory and technology for the nonlinear and multidimensional traversal of complex subject matter. In D. Nix & J.S. Rand (Eds), *Cognition, Education and Multimedia: Exploring Ideas in High Technology*. pp. 163-205. Hillsdale, NJ: Lawrence Erlbaum Associates.

Stake, R. E. (1994). Case studies. In: N. K. Denzin, & Y. S. Lincoln, (Eds), *Handbook of Qualitative Research* (pp. 236-247). Thousand Oaks: Sage.

Stake, R. E. (1995). *The Art of Case Study Research*. Thousand Oaks: Sage.

Suchman, L. (1987) *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge: Cambridge University Press.

Taveira, M. (1991). Hypermedia and Career Counselling for Personal Flexibility. *Canadian Journal of Counselling*, 25, 136-145.

Taylor, C. & Self, J. (1990). Monitoring hypertext users. *Interacting with Computers*, 2, 297-312.

Thimbleby, H. (1990). *User Interface Design*. Wokingham: Addison-Wesley.

Thuring, M., Hannemann, J. & Haake, J.M. (1995) Hypermedia and cognition: designing for comprehension. *Communications of the ACM*, August, 38(8), 57-73.

Tognazzini, B. (1992). *Tog on Interface*. Reading MA: Addison-Wesley.

Tolhurst, D. (1992). A Checklist for Evaluating Content-Based Hypertext Computer Software. *Educational Technology*, 32, 17-21.

Trauth, E. M. & O'Connor, B. (1991). A study of the interaction between information technology and society: an illustration of combined qualitative research methods. In H. E. Nissen, H.K. Klein & R. Hirschheim, (Eds), *Information Systems Research: Contemporary Approaches and Emergent Traditions* (pp. 131-144). Amsterdam: Elsevier (North-Holland).

Travis, D. (1991). *Effective Color Displays*. London: Harcourt Brace Jovanovich.

Trenouth, J. (1991). A survey of exploratory software development. *The Computer Journal*, 34 (2), 153-163.

Trigg, R. (1990) Link anchoring in dynamic media: A user interface challenge. In N. Streitz (panel proposer), *What's specific about User-Interfaces for Hypertext Systems?* (pp 359-361) In A. Rizk, N. Streitz & J. Andre, (Eds), *Hypertext: Concepts, Systems & Applications* (pp. 354-361). Cambridge: Cambridge University Press.

Tullis, T.S. (1988). Screen design. In M. Helander (ed.), Handbook of Human-Computer Interaction pp. 377- . Amsterdam: North-Holland.

Van Vliet, P. A. & Wilson, R. (1992). A hypertext development methodology. *Journal of Database Management*, 4(2), 18-29.

Wadlow, M.G. (1990). The Andrew system: The role of human interface guidelines in the design of multimedia applications. Special Issue: Hypermedia and artificial intelligence. *Current Psychology Research and Reviews*, (9), 181-191.

Waern, Y. (1989). *Cognitive Aspects of Computer Supported Tasks*. Chichester: John Wiley & Sons.

Waterworth, J. & Chignell, M. (1991). A model for information exploration. *Hypermedia*, 3(1), 35-58.

Waterworth, J. A. (1992). *Multimedia Interaction with Computers: Human Factors Issues*. NY: Ellis Horwood.

Waterworth, J. A., Chignell, M H. & Zhai, S. M. (1993). From icons to interface models: designing hypermedia from the bottom up. *International Journal Of Man-Machine Studies, 39*, 453-472.

Weibel, S. (1989). Automated document architecture processing and tagging (ADAPT). *Annual Review of OCLC Research*, 4-6.

Weick, C. (1984). Theoretical assumptions and research methodology selection. In F.W. McFarlan (Ed.). *The Information Systems Research Challenge - Proceedings* (pp. 113-132). Boston: Harvard Business School Press.

Weinberg, G. (1971). *The Psychology of Computer Programming*. New York: Van Nostrand Reinhold.

Wertheimer, M. (1959) *Productive Thinking* (enlarged edition), NY: Harper & Row.

Whitby, Z. (1992). *Promotional Publications: A Guide for Editors*. Leeds: Heist.

Whitefield, A. (1989) Constructing appropriate models of computer users: the case of engineering designers. In J. Long & A. Whitefield (Eds), *Cognitive Ergonomics and Human-Computer Interaction*. Cambridge: Cambridge University Press (66-94)

Whiteside, J., Bennett, J. & Holzblatt, K. (1988). Usability engineering: our experience and evolution. In M. Helander (Ed.), *Handbook of Human-computer Interaction* (pp. 791-817). Amsterdam: Elsevier (North-Holland).

Williams, F. R., Ronald E; Rogers, Everett M (1988). *Research Methods and the New Media*. New York: The Free Press (MacMillan).

Williams, N. (1992). New technology, new writing... new problems? In P. O. Holt & N. Williams (Eds), *Computers and writing: State of the art*. Oxford: Intellect.

Winograd, T. & Flores, F. (1986). *Understanding Computers and Cognition: A New Foundation for Design*. Norwood, NJ: Ablex.

Wittgenstein, L. (1978). *Philosophical Investigations*. (Trans. G.E.M. Anscombe.) Oxford: Blackwell.

Wixon, D., Holtzblatt, K., & Knox, S. (1990). Contextual design: an emergent view of system design. In *Proceedings of the ACM CHI'90 Conference (1990) Human Factors in Computing Systems*, 1-5 April. (pp. 329-336) Seattle, WA.

Wright, P. (1981) Informed design for forms. *Information Design Journal*. 2(3/4), (151-178).

Wright, P. (1988) Functional literacy: reading and writing at work. *Ergonomics*, 31(3) 265-290

Wright, P. (1989). Interface alternatives for hypertexts. *Hypermedia*, 1(2), 146-166.

Wright, P. (1990). Hypertext as an interface for learners: some human factors issues. In D.H. Jonassen & H. Mandl (Eds), *Designing Hypermedia For Learning* (pp. 169-184). Berlin: Springer-Verlag.

Wright, P. (1990). It's what you can do, and the way you can do it. In N. Streitz (panel proposer), *What's specific about User-Interfaces for Hypertext Systems?* (pp 358-359). In A. Rizk, , N. Streitz & J. Andre, (Eds), *Hypertext: Concepts, Systems & Applications* (pp. 354-361). Cambridge: Cambridge University Press.

Wright, Patricia and Lickorish, Ann (1990). An empirical comparison of two navigation systems for two hypertexts. In: R. MacAleese, and C. Green, (Eds), *Hypertext: State of the Art* (pp. 84-93). Oxford: Intellect.

Yankelovich, N., Meyrowitz, N., & van Dam, A. (1985). Reading & writing the electronic book. *IEEE Computer*, (October), 15-30.

Yin, R. K. (1994). *Case study research: Design and methods* (2nd ed.). Thousand Oaks: Sage.

Zuber-Skerritt, O. (Ed.). (1991). *Action Research for Change and Development*. Aldershot: Avebury.

Appendix 1

These are the principles and guidelines which formed the raw data for the taxonomy of guidelines described in Chapter 4, sorted into the Feature. and Procedure categories. The author is given, followed by the source discipline or subject area, level, either principle or guideline, and allocated domain or scope category. In the case of feature, guidelines, this refers to the domain. In the case of procedure guidelines, it refers to the scope. Numbers given are from the original database, therefore are not continuous in each set. It should be noted that this material was exported from a database of these guidelines, and whilst the material has been checked against the originals, it should not be considered a complete substitute for the originals, which should be referred to for guaranteed accuracy. Some of the material contains its own references. These are not cited in the bibliography, and readers are again directed to the original sources.

1a: Feature guidelines

1 [The design principles applicable to hypertext are:] Consistency: the task should be structured so that a consistent presentation is used and consistent sequences of actions are required in similar situations. *Hardman & Sharratt* 1990. Hypertext/principle/user interface.

2 Mental processing: the hypertext should not complicate the reader's information-gathering tasks and impose excessive mental processing requirements. The author should be trying to (i) minimise the overall mental load by reducing the requirement for readers to remember the objects, actions, codes and abbreviations they are working with' (ii) minimise the task-specific mental processing by arranging for efficient completion of typical reader tasks. *Hardman & Sharratt* 1990. Hypertext/principle/user interface.

3 Ease of learning and use. The aim is to reach a suitable balance between ease of learning and ease of use. Ease of learning focuses on enabling the reader to become proficient with the hypertext with minimal training and practice. Ease of use is achieved by minimising the steps or actions taken by knowledgeable readers. *Hardman & Sharratt* 1990. Hypertext/principle/user interface.

4 Flexibility. The hypertext should be capable of adaptation to the needs of the user. This principle covers the need to (i) design the hypertext for different types of users and levels of experience; and (ii) provide multiple paths that allow readers to bypass certain parts of the hypertext. *Hardman & Sharratt,* 1990. Hypertext/principle/user interface.

5 In addition one further principle should be considered for certain hypertexts: Task compatibility. This embodies a number of related concepts with the author having to ensure that: (i) there is stimulus-response compatibility - where the hypertext information is presented in a form that is suitable for the reader's tasks; (ii) the author takes advantage of any physical analogies that aid the presentation of and navigation between information; and (iii) the layout and coding applied to information conforms to reader's expectations. *Hardman & Sharratt* 1990. Hypertext/principle/user interface.

6 Display of link (S) When reading a hypertext the data input from the reader is restricted to selecting and actioning a link. To help the reader with choosing a link the positions of the links on the screen should be obvious. Highlighting the links in some way would satisfy this but, with a large number of links, the screen may look cluttered. Some conventions, for example defined areas containing links, can be adopted for hypertexts where readers are likely to become expert users. *Hardman & Sharratt,* 1990. Hypertext/guideline/user action - link display.

7 Large pointing area for option selection (S). Where there are small linked items in a hypertext the active area for the link can be larger than the visual representation. [Refers to Fig 1 - not included here.] *Hardman & Sharratt* 1990. Hypertext/guideline/user action - link display

8 Identify link types (S-D). Different styles of highlighting and graphical information can be used to distinguish different types of link. This is especially important when links are embedded in text> For example a link that brings up a small note might be one style, and a link that takes the reader to the middle of a completely different section of the hypertext another style. When selecting which types to visual code to use care should be taken to keep the number small so the reader is not overloaded. *Hardman & Sharratt,* 1990. Hypertext/guideline/user action - link display.

9 Highlighting critical information (S). Certain parts of the information contained in the hypertext may be particularly important. This information can be highlighted but needs to be a different style from those used to denote the links in the hypertext. Highlighting should only be used for a small proportion of the information on the screen. *Hardman & Sharratt*, 1990. Hypertext/guideline/information display - links.

10 Assignment of visual codes (S). Forms of visual coding that can be used are text style, graphical style, colour, brightness, flashing or some combination of these features. Flashing should be used with great care since it can be effective the first time round but becomes irritating with frequent use. Colour should be used as a redundant feature, (i.e. colour is paired with another form of coding), an preferably with options to allow readers to select the colours they want for each code. When using graphical representations extra care is required to produce clear codings of link types and critical information. [refers to fig 2 - not included here]. *Hardman & Sharratt*, 1990. Interface design/Guideline/information display - visual codes.

11 Display necessary information (S). The items a reader requires on screen are:
(i) a clear title;
(ii) the information the reader is interested in;
(iii) indications of where the links are in this material;
(iv) other links to known places in the hypertext;
(v) sufficient context information to inform readers where they currently are (see the following section for the context information that should be provided). *Hardman & Sharratt*, 1990. Hypertext/guideline/information display.

12 Grouping information (S). Information should be arranged to make relationships clear. This is particularly important for graphical information. Where possible, different windows (or areas of the screen) should be used for different types of information. *Hardman & Sharratt*, 1990. Hypertext/guideline/information display.

13 Ordering information (S). Ordering of lists should be designed to assist readers' tasks, and in hypertext this can be enhanced by having multiple orderings of the same information. As well as this, what would traditionally be a linear list can be broken down into a hierarchical structure containing shorter sublists. *Hardman & Sharratt*, 1990. Hypertext/guideline/information display.

14 Context for displayed information (D). Readers should not be required to remember where they are when they arrive at a new screen, after actioning a link, there should be sufficient information to reorient themselves. Information on what the current section is, and where in the current section the reader is should be either immediately available on the screen, or directly accessible by a mechanism made known to the reader. *Hardman & Sharratt*, 1990. Hypertext/guideline/dialogue design.

15 Reading extended information (D). Where immediately relevant information takes up more than one screen the reader should be able to move easily between the relevant displays. Some hypertext systems allow scrolling, but readers should not be required to scroll too far or they may lose their sense of where they are. *Hardman & Sharratt*, 1990. Hypertext/guideline/dialogue design.

16 Terminology and wording (S). Any wording the author uses to guide the reader should follow standard guidelines:
(i) terminology should be familiar (or there should be easy access to definitions) and consistent;
(ii) abbreviations should be explained;
(iii) sentence structure should be simple;
(iv) instructions should be affirmative and in the active voice. *Hardman & Sharratt*, 1990. Hypertext/guideline/dialogue design.

17 Effective use of graphics (S). Graphics can be used in a number of ways, other than illustrating the material the reader is interested in. Icons can be used for frequently occurring links; diagrams of nodes and links can be used to help orient the reader (this type of map need not contain every link, but main links between sections). Icons should be clear and legible and standardised throughout the hypertext. *Hardman & Sharratt*, 1990. Hypertext/guideline/graphics design.

18 Consistent formats (D). The layout of material across different screens should remain as consistent as possible. For example, keep the contents and help section buttons always in the same place, arrange similar types of information in similar ways. Fonts and styles should be used consistently throughout the hypertext. *Hardman & Sharratt*, 1990. Hypertext/guideline/dialogue design.

19 Help on using links (S-D). The hypertext author only has to deal with a small part of aiding the reader since the hypertext system itself should take care of many of the reader's requirements. The "errors" a reader is likely to make are (i) clicking on an item which is not linked; or (ii) actioning the wrong link. In

the first case some hypertext systems allow the author to display a message such as "Click on a surname to see the numbers" in the second case some hypertexts systems allow readers to backtrack to where they just came from, otherwise the author can include links to take the reader back. *Hardman & Sharratt*, 1990. Hypertext/guideline/on-line assistance.

20 Help always available (S). Help at a general level and help specific to the reader's current position should always be available and obtained through a standard procedure, e.g. a help icon is always displayed on the screen. *Hardman & Sharratt*, 1990. Hypertext/guideline/on-line assistance.

21 The following guideline applies only to hypertexts where there is a high proportion of link following compared with reading the information.

Minimise cursor movement (S-D). Arrange items to minimise the distance the reader has to travel to reach the links. *Hardman & Sharratt*, 1990. Hypertext/guideline /dialogue design.

22 '...just enough up front-details on demand.' This paradigm suggests that information overload can be minimised by providing just enough information initially to ensure comprehension and providing details on demand to the user. *Kreitzberg* 1989. Hypertext/principle /information display.

23 '....that the amount of information presented at each stage should be "just enough". "Just enough means that the presentation should be complete but not elaborated." *Kreitzberg* 1989. Hypertext/principle/information display.

24 The third element specifies that detail be provide on demand. This means that subordinate information should be removed from the initial presentation and accessed by means of links. Wherever there is a need for elaboration a link should be available so that the reader can obtain the details at the point they are mentioned. However, care should be taken not to clutter up the screen with redundant links. Our style is to limit the number of links per screen to 6-8. *Kreitzberg* 1989. Hypertext/principle /information design - display.

25 1. Identify the domain of discourse so that a context is established. Meaningfulness is highly dependent on context (Bransford and Johnson, 1972). For example, the phrase "I want to press a suit" has a very different meaning depending on whether you are talking to a lawyer or a tailor. *Kreitzberg* 1989. Hypertext-HCI/Guideline/Information design.

26 2. Identify pre-existing knowledge which the individual needs for complete understanding. Make such knowledge available through links so that the reader can access it and process the remainder of the information meaningfully. *Kreitzberg* 1989. Hypertext-HCI/guideline/information content.

27 3. Present the information in logical sequence so that the information which came before provides context for that which follows. *Kreitzberg* 1989. Hypertext/guideline/information design - content.

28 4. Identify details which the individual can obtain to elaborate the information in the initial presentation. When constructing a detail level, apply these same rules to it. *Kreitzberg* 1989. Hypertext/guideline/information design - content.

29 Design Goal 1: Simplicity of design elements. *Brooks* 1993. Technical communication/principle/general.

30 Hypermedia design tip 1: keep design elements to a minimum. Brooks 1993. Technical communication/principle/general.

31 Hypermedia design tip 2: Repeat the placement and format of major document-wide elements such as headings, text blocks, and navigational controls. *Brooks* 1993. Technical communication/guideline/user interface.

32 Hypermedia design tip 3: Avoid piling up special effects, so that those used will retain their attention-getting function. *Brooks* 1993. Technical communication/guideline/user interface.

34 Hypermedia design tip 4: Keep design elements as well as content in an appropriate tone for the sponsoring publisher of the on-line document. Appropriateness depends on the intended user and the sponsoring publisher. *Brooks* 1993. Technical communication/principle/general design features.

35 Hypermedia design tip 5: Do not simply dump word processing files or screens of spreadsheets on-line. *Brooks* 1993. Technical communication/guideline/information design.

36 Design Goal 3: Function of the hypermedia document as an aid to information access. Function refers to how the hypermedia document will be used. Hypermedia publications have a wide variety of purposes and uses, but the key design consideration is how to provide access to stored information or multimedia events. *Brooks* 1993. Technical communication/principle/information access-retrieval.

37 Hypermedia design tip 6: Provide the user with multiple means of controlling his or her navigation through the document to encourage more than browsing or watching. *Brooks* 1993. Technical communication/guideline/information access & retrieval, navigation.

38 Hypermedia design tip 7: Provide help explaining or showing how the hypermedia publication functions as soon as the user opens the document, and maintain access to help throughout the user's interaction. Give the user a forecast of how the hypermedia publication interaction will work. *Brooks* 1993. Technical communication/guideline/on-line help.

45 4. Create a good attract mode. CAPS competes with other attractions in public space. The attract mode of the software must be appealing enough to get the user to approach the system without giving false messages as to the nature of the application. *Heller* 1991. Interactive multimedia - computing access in public spaces/guideline/user interface

46 5. Plan for high usage. Public space computing requires that the application get the message across quickly and easily so that as many users as possible can use it. *Heller* 1991. Interactive multimedia - computing access in public spaces/principle/user interface

47 Help should be available to tell or show the user what specific input selections mean as well as give clear and concise error messages. *Heller* 1991. Interactive multimedia - computing access in public spaces/Guideline/User interface.

49 6. Use content related metaphors. Metaphors used to create icons and other graphical cues must come from the content space and be easily understandable by, and not offensive to, a variety of users. (Sukaviraya and Moran, 1990; del Galdo 1990). *Heller* 1991. Interactive multimedia - computing access in public spaces/guideline/user interface.

50 7. Use direct manipulation. The use of direct manipulation allows the user to interact with the application through the least amount of abstraction. In addition, direct manipulation reduces the number of mechanical input devices attached to the system. *Heller* 1991. Interactive multimedia: computing access in public spaces/guideline/user interface.

51 8. Use multiple or redundant modalities. CAPS is available to a broad spectrum of users, each with their own learning style. Multiple forms of the same information allow the user to learn in the style most comfortable for them. In addition redundant modalities insure the clarity of instructions and information. *Heller* 1991. Interactive multimedia: computing access in public spaces/guideline/user interface.

52 9. Provide quick and easy exits. Because CAPS users are often pressed for time or without deep commitment to an application, the user has to be able to escape from the application at any point. This escape can be a return to the beginning of the program or to leave the application entirely. *Heller*,1991. Interactive multimedia: computing access in public spaces/guideline/user interface.

53 10. Use few or no mechanical input devices. This type of equipment requires repair and is subject to theft and vandalism. The lack of on-site support staff as well as the openness of CAPS is the basis of this guideline. *Heller* 1991. Interactive multimedia: computing access in public spaces/guideline/human-computer interface - hardware.

73 4. Respect chunking. The information to be presented needs to be organized into small "chunks" that deal with one topic, theme or idea. Chunks may be 100 words or 1000 words but when a chunk reaches 10 000 words the author should consider restructuring into multiple smaller chunks. Screens are still usually small and hard to read, so lengthy linear texts are not as pleasant. Each chunk represents a node or document in the database. *Shneiderman* 1989 General hypertext/principle./information content - user interface.

74 5. Show inter-relationships: Each document should contain links to other documents. The more links contained in the documents, the richer the connectivity of the hypertext. Too few links means that the medium of the hypertext may be inappropriate, too many links can overwhelm and distract the reader. Author preferences range from those who like to put in a maximum of one or two links per screen, to the more common range of two to eight links per screen, to the extremes of dozens of selectable links per screen. *Shneiderman* 1989. General hypertext/principle/linking.

76 8. Ensure simplicity in traversal: Authors should design the link structure so that navigation is simple, intuitive, and consistent throughout the system. Movement through the system should be effortless and require a minimum of conscious thought. Find simple, comprehensible, and global structures that the readers can use as a cognitive map. Be sensitive to the possibility that the user will get "lost in hyperspace" and develop the system so recovery is simple. *Shneiderman* 1989. General hypertext/principle/linking/navigation.

77 9. Design each screen carefully- Screens should be designed so they can be grasped easily. The focus of attention should be made clear, headings should guide the reader. Visual layout is very important in screen design. *Shneiderman* 1989 p125. general hypertext/guideline/user interface - screen design.

78 10. Require low cognitive load - minimize the burden on the user's short-term memory. Do not require the user to remember things from one screen to another. The goal is to enable users to concentrate on their tasks and the contents while the computer vanishes. *Shneiderman* 1989. General hypertext/principle/user interface.

79 Rule 1. The very existence of links in hypermedia conditions the reader to expect purposeful, important relationships between linked material. *Landow* 1991. Educational hypertext/principle/links.

80 Rule 2. The emphasis on linking materials in hypermedia stimulates and encourages habits of relational thinking in the reader. *Landow* 1991. Educational hypertext/principle/links.

81 Rule 3. Since hypermedia systems predispose users to expect such significant relationships among documents, those documents that disappoint these expectations must appear particularly incoherent and nonsignificant. When users follow links and encounter materials that do not appear to possess a significant relation to the document form which the link pathway originated, they feel confused and resentful. *Landow* 1991. Educational hypertext/principle/links.

82 Rule 4. The author of hypermedia materials must provide devices that stimulate the reader to think and explore them. *LLandow* 1991. Educational hypertext/principle/browsing.

83 Rule 5. The author of hypermedia must employ stylistic devices that permit readers to navigate materials easily and enjoyably. *Landow* 1991. Educational hypertext/principle/navigation.

84 Rule 6. Devices of orientation permit readers (a) to determine their present location, (b) to have some idea of that location's relation to other materials, (c) to return to their starting point, and (d) to explore materials not directly linked to those in which they presently find themselves. *Landow* 1991. Educational hypertext/principle/navigation.

85 Rule 7. Authors should consider employing several overviews to organize the same body of material and to assist readers to gain easy access to it. *Landow* 1991. Educational hypertext/principle/navigation - overviews.

86 Rule 8. Never place link markers independent of accompanying text or image. *Landow* 1991. Educational hypertext/principle/links.

87 Rule 9. When creating a link or positioning a link, remember that all links are bidirectional. Some systems, like HyperCard, only permit uni-directional linking. *Landow* 1991. Educational hypertext/guideline/links.

88 Rule 10. Avoid linking to words or phrases that only provide appropriate points of arrival but give the reader no suggestion of where the link might lead. *Landow* 1991. Educational hypertext/guideline/links.

89 Rule 11. Place the link marker in close proximity to a text that indicates the probable nature of the link destination. *Landow* 1991. Educational hypertext/guideline/links.

90 Rule 12. When creating documents, assist readers by phrasing statements or posing questions that provide obvious occasions for following links.*Landow* 1991. Educational hypertext/principle/links.

91 Rule 13. When possible provide specific information about a link destination by directly drawing attention to it. *Landow* 1991. Educational hypertext/guideline/links.

92 Rule 14. Linked graphic materials must appear with appended texts that enable the user to establish a relation between a point of departure and that of arrival. ....From this rule follows two more: [below]. *Landow* 1991. Educational hypertext/principle/nodes.

93 Rule 15. The entire text accompanying visual materials and not just the opening sentence or two serves as an introduction. And: [below] *Landow* 1991. Educational hypertext/guideline/nodes.

94 Rule 16. The text accompanying an image does not have to specify all relevant information the author wishes the reader to have: rather emphasising that a relationship exists at all may be enough. From which follows: [below]. *Landow* 1991. Educational hypertext/guideline/nodes.

95 Rule 17. Texts serve not only to provide information but also to reassure the reader that the link embodies a significant relationship and to provide some hint, however incomplete, of how that relationship can be formulated by the reader. *Landow* 1991. Educational hypertext/principle/nodes.

96 Rule 18. When creating documents for hypermedia, conceive the text units as brief passages in order to take maximum advantage of the linking capabilities of hypermedia. *Landow* 1991. Educational hypertext/guideline/nodes.

103 3. Decide how to present the subject matter to your users. *HyperCard Stack Design Guidelines* 1989. Hypertext: HyperCard/guideline/information presentation.

104 Guideline 4. Make your stack easy to navigate.

Navigation, the part of the stack's user interface by which users move around in the stack, is the most important component of stack design. If users are confused or frustrated when trying to move around in a stack, they will quit, no matter how useful the subject matter. Whatever its design, a stack's navigation system must address five user needs:
• Context: What's in this stack?
Location: Where am I now within the stack?
• Destination choices: Where can I go?
• Travel methods: How do I get there?
• Progress indicators: Where have I already been? *HyperCard Stack Design Guidelines,* 1989. Hypertext: HyperCard/guideline/user interface.
105 Guideline 5: Introduce people to your stack. *HyperCard Stack Design Guidelines* 1989. Hypertext: HyperCard/guideline/user interface.
106 Make your stack simple to use, so minimal learning is required. *HyperCard Stack Design Guidelines* 1989. Hypertext: HyperCard/guideline/user interface.
107 Include a title card that describes a stack's purpose. *HyperCard Stack Design Guidelines,* 1989. Hypertext: HyperCard/guideline/user interface.
108 Get the user doing something quickly in the stack, such as clicking a button or typing a name. *HyperCard Stack Design Guidelines* 1989. Hypertext: HyperCard/guideline/user interface.
109 Provide an explicit introduction to the stack. Give information in small chunks, so users don't have to assimilate too much at once. *HyperCard Stack Design Guidelines,*1989. Hypertext: HyperCard/guideline/user interface.
110 Don't keep secrets from users: tell them how your stack works. *HyperCard Stack Design Guidelines,*1989. Hypertext: HyperCard/guideline/user interface.
111 Provide specific help for your stack - not just links to the HyperCard Help system - with a button that's always available. *HyperCard Stack Design Guidelines,*1989. Hypertext: HyperCard. Subsidiary guideline/guideline/help facility.
112 Put buttons in the help system so uses can find what they need without reading through long screens of information. *HyperCard Stack Design Guidelines,*1989. Hypertext: HyperCard/guideline/help facility.
113 Use graphics, animation, and sound to illustrate the help concepts. *HyperCard Stack Design Guidelines* 1989. Hypertext: HyperCard/guideline/help facility.
114 Explain your stack's buttons and remind users how to move round the stack. *HyperCard Stack Design Guidelines,*1989. Hypertext: HyperCard/guideline/help facility.
115 When designing your help system, follow the same principles that you use to design an effective stack. *HyperCard Stack Design Guidelines,*1989. Hypertext: HyperCard/guideline/help facility.
116 Guideline 6. Integrate text, graphic design and audio design. *HyperCard Stack Design Guidelines* 1989. Hypertext: HyperCard/guideline/user interface.
117 Design consistent card and background layouts. *HyperCard Stack Design Guidelines* ,1989. Hypertext: HyperCard/guideline/user interface.
119 Provide high-quality graphic and audio design. *HyperCard Stack Design Guidelines,*1989. Hypertext: HyperCard/guideline/graphics & audio.
120 Incorporate the general design of Apple's Human Interface Guidelines. *HyperCard Stack Design Guidelines* 1989. Hypertext: HyperCard/guideline/user interface.
125 Details make the difference. Your stack should have
• no typographical or grammatical errors
• text and graphics that relate to each other (no mismatches)
• buttons that all work
• scripts that all run without crashing and that produce the intended result
• logical presentation sequence to cards (so that the Show all cards command produces orderly result
• alert and dialog boxes that all work (check all options, including Cancel)
• navigation and links that all work and are connected to the correct destinations
• consistency throughout the stack
• no glitches with sound or animation timing on any Macintosh model
Check to make sure that everything's linked, that users can't get stuck in a dead end, and that scripts don't bomb. Check that copyright and legal notices have been inserted. Verify that no viruses exist on the master disk.

When you have a final disk with no bugs, put everything associated with this project . . . . into order while they're fresh in your memory and save them. *HyperCard Stack Design Guidelines*, 1989. Hypertext: HyperCard/guideline/general.

126 Metaphors from the real world

• Use concrete metaphors and make them plain, so that users have a set of expectations to apply to computer environments.

• Whenever appropriate, use audio and visual effects that support the metaphor. *HyperCard Stack Design Guidelines* 1989. Hypertext: HyperCard/principle/user interface.

127 Direct manipulation

• Users want to feel they are in charge of the computer's activities.

• Tell users their options by providing visible choices, ways to make their choices, and feedback acknowledging their choices. *HyperCard Stack Design Guidelines* 1989. Hypertext: HyperCard. principle/user interface.

128 See-and-point (instead of remember-and-type)

• Users select actions from alternatives presented on the screen.

• Users rely on recognition, not recall; they shouldn't have to remember anything the computer already knows.

• Most programmers have no trouble working with interfaces that require memorisation. The average user is not a programmer. *HyperCard Stack Design Guidelines* 1989. Hypertext: HyperCard/principle/user interface.

129 Consistency

Effective applications are both consistent with themselves and consistent with one another. Consistency within a stack is essential. The look, the usage, and the stack behavior should be the same throughout. The way the user does things should always be consistent within a stack. For example, your stack should have a consistent design for these elements:

• graphic look
• grouping of buttons
• placement of buttons
• visual and audio feedback
• card layout
• background for cards with similar functions
• stack structure

HyperCard Stack Design Guidelines, 1989 p179. Hypertext: HyperCard/principle/user interface.

130 WYSIWYG (what you see is what you get)

• There should be no secrets from the user, no abstract commands that only promise future results.

• There should be no significant difference between what the user sees on the screen and what eventually gets printed. *HyperCard Stack Design Guidelines* 1989. Hypertext: HyperCard/principle/user interface.

131 User control

The user, not the computer, initiates and controls all actions....... *HyperCard Stack Design Guidelines*, 1989. Hypertext: HyperCard/principle/user interface.

132 Feedback and dialog

• Keep the user informed
• Provide immediate feedback
• Make user activities simple at any moment, though they may be complex taken together. *HyperCard Stack Design Guidelines* 1989. Hypertext: HyperCard/principle/user interface.

133 Forgiveness

• Users make mistakes; forgive them.
• The user's actions are generally reversible - let the user know about any that aren't.
• Users get lost in stacks; help them find their way. *HyperCard Stack Design Guidelines* 1989. Hypertext: HyperCard stacks/principle/user interface.

134 Perceived stability

• Users feel comfortable in an a computer environment that remains understandable and familiar rather than changing randomly. *HyperCard Stack Design Guidelines* 1989. Hypertext: HyperCard/principle/user interface.

135 Aesthetic integrity
• Visually confusing or unattractive displays detract from the effectiveness of human-computer interaction.
• Different "things" should look different on the screen.
• Messes are acceptable only if the user makes them - stacks aren't allowed this freedom. *HyperCard Stack Design Guidelines* 1989. Hypertext: HyperCard stacks/principle/user interface.
136 Design for monochrome first. *Travis* 1991. Computer science: visual displays/principle/user interface - screen design-colour.
137 Use the user's model. *Travis* 1991. Computer science: visual displays/principle/user interface/screen design - colour.
138 1: Use colour to group *Travis* 1991. Computer science: visual displays/principle/user interface - screen design - colour.
139 2: Use colour for emphasis. *Travis* 1991. Computer science: visual displays/principle/user interface - screen design - colour.
140 Choose a dark or dim background, such as deep blue on a CRT, and bright foreground colors. *Travis* 1991. Computer science: visual displays/guideline/user interface/screen design/colour.
141 Relate separate areas by using a common hue, for example a common background or text color. *Travis* 1991. Computer science: visual displays/guideline/user interface - screen design- colour.
142 Highlight regions by using colors that contrast with the background in hue, saturation and luminance. *Travis* 1991. Computer science: visual displays/guideline/user interface - screen design - colour.
143 Be careful not to highlight too many groups at once. Restrict highlighting to one or two key items. *Travis* 1991. Computer science: visual displays/guideline/user interface - screen design - colour.
144 Question whether color coding is the best way to represent the information. There are numerous methods of coding information: is color the best for this particular application? *Travis* 1991.
Computer science: visual displays/guideline/user interface - screen design - colour.
145 The colours used in a coding task should be:
Discriminable: the chosen colors should be discriminable from each other by the predicted group of users. *Travis* 1991. Computer science: visual displays/guideline/user interface - screen design - color coding.
146 [The colours used in a coding task should be:]
Detectable: the chosen colors should be detectable by the predicted group of users under the ambient illumination that the display will be used. [sic]. *Travis* 1991. Computer science: visual displays/guideline?/user interface - screen design - color coding.
147 [The colours used in a coding task should be:]
Perceptually equal steps: the discrimination steps between the chosen colors should appear approximately the same. *Travis* 1991. Computer science: visual displays/guideline?/user interface - screen design - color coding.
148 [The colours used in a coding task should be:]
Meaningful: the code should have some relevance to the user. *Travis* 1991. Computer science: visual displays/guideline?/user interface - screen design - color coding.
149 [The colours used in a coding task should be:]
Consistent: if the 'Help' option is coded green on one screen, it should be coded green on all screens. Moreover, this color should be 'reserved' once it has been claimed for an option. *Travis* 1991. Computer science: visual displays/guideline?/user interface - screen design - color coding.
150 [The colours used in a coding task should be:]
Aesthetically pleasing: use as few colors as possible and ensure that the chosen colors do not clash or vibrate. *Travis* 1991. Computer science: visual displays/guideline/user interface - screen design - color coding.
197 Apply hypertext with precision and restraint. Keep the design simple. Elaborate hypertext systems are hard to learn and tricky to operate. Keep the system simple, or at least make it appear simple to the user. *Horton* 1990. On-line documentation/guideline/user interface.
199 Give users a clear starting point.
Designate one topic in the hypertext as the root or home topic. This topic serves as the starting point for the novice users (*Shneiderman* 1989). Design this topic to introduce the document, show how it is organised, and provide convenient access to key topics. To let the user jump to key topics:
• Include a table of contents for the document

• Include an alphabetical index of key topics.
• Make the topic the top level of a hierarchical menu leading to key topics.
Horton 1990. On-line documentation/guideline/user interface.

200 Link moderately.

User links to show interrelationships. Using too few links defeats the purpose of creating a hypertext and questions whether the document is properly represented by hypertext. Too many links may only confuse and distract the user. The proper number of links depends on the subject, but two to eight links per display seems a reasonable number (*Shneiderman* 1989). *Horton* 1990. On-line documentation/guideline/linking.

202 Display organisation visually

Because hypertext systems allow arbitrary and ad hoc structures, it is important that they make the organisation explicit and clear. Many systems display a map of the hypertext and let the user navigate by selecting nodes from this map. *Horton* 1990. On-line documentation/principle/user interface & navigation.

204 The primary goal.

It is the job of the author of the hyperdocument to communicate information in such a way as to make it as valuable as possible to the reader. Any other goal is subordinate to this..... To create value for the reader the hyperdocument must have maximum clarity and, like a smoothly oiled machine, must help the reader to navigate to the items which help him.

Maximum clarity requires

• Clear structuring
• Clear organisation of ideas
• Clear English
• Clear diagrams
• Clear navigation

Martin 1990. Hypertext/principle/guideline/general.

210 3-D effects

Try to avoid different light and shadow effects. don't use 3-D buttons and several layers of dialog boxes and function bars. A 14" screen is small and these features waste a lot of space by additional lines indicating shadows. Again, a side effect is the load on the user which he has to carry by understanding the different layers of objects. And from the artist's point of view, the change of light sources that takes place when buttons are pressed is confusing/inconsistent anyway! *Meusel, Eickmeyer & Koslowski* 1995. Hypertext for on-line help/guideline/user interface - screen design.

211 Typography

Use one or two font styles, not more ('fontitis'). Avoid visual rivers (Kahn 1992) by using 'align left' instead on 'justified' when formatting your paragraphs. Snap the text-portions to the grid, and use margins and white space to improve readability. Use specific fonts for the screen: Arial or Helvetica, for example. Don't use fonts like Times Roman. They are nice to read on paper, but the readability is bad on screen. Remember: You're designing information for the computer screen (Kahn, Lenk 1992). [More]. *Meusel, Eickmeyer & Koslowski* 1995. Hypertext for on-line help/guideline/user interface & information design.

212 Colors

Use a few colors only! Also, make sure that objects with the same color have the same function or semantics. Plain white pages can be more attracting than fancy colors. Use ideas from color theory for defining good contrasts and avoiding disharmony. Remember: Apple Macintosh systems mostly use black and white interfaces, which can be much more professional and readable than badly-designed color applications.

Also be aware of technical problems: Colors used by application software like ToolBook will look different on different displays. For example, we experienced that 'petrol' defined on a PC changed to 'light blue' on a notebook. *Meusel, Eickmeyer & Koslowski* 1995. Hypertext for on-line help/guideline/user interface - screen design - colour.

213 Orientation and Navigation

Allocate big parts of the screen for orientation bars (headlines, names of structural elements) and navigation facilities (Meusel, Schröcksnadel, Schiff 1993). Good ideas for 'navigation' concepts are offered by Smalltalk class hierarchy browsers and by the Norton Commander. Combine these concepts with the common 'hotword' approach. Hotwords can be activated fast, typically by a single mouse click. But sometimes the link anchored in a hotword doesn't take the reader to the expected place, because of bad wording, for example. Therefore we recommend the additional use of a navigation bar. See (Meusel, Schröcksnadel, Schiff 1993) for detailed discussion. *Meusel, Eickmeyer & Koslowski* 1995. Hypertext for on-line help/guideline/orientation & navigation.

214 Structured skeleton beyond the visible interface

If you dive beyond the interface, a clear structure should become visible. A professional visual design is useless without a good 'engine' (HDM (Garzotto, Paolini, Schwabe, 1991), for example). Figure 1 [not included] shows the combination of the last two issues. In an electronic service manual, two main information types are defined ('facts' for look-up and 'procedures' for performing service tasks), which are offered in the navigation bar. By selecting one of them, the next level of the hierarchy becomes visible. To improve readability and usability of the hyperdocument, hotwords are offered in a margin and the information type of the link destination is always added to the hotword. This is also a simple way of visualizing link types that are defined on conceptual level. *Meusel, Eickmeyer & Koslowski* 1995. Hypertext for on-line help/guideline/information structure?

219 How to create a good topic:

1. Make topics coherent. Effective topics are coherent and clearly focused throughout. The benchmark of a good topic is that it:

• Answers one question
• About one subject
• For one purpose . *Horton* 1994. On-line documentation/principle/nodes.

220 How to create a good topic:

2. Craft recombinant building blocks:
- make topics self-contained
Make each independently accessible topic of the document complete in itself.
- keep topics rhetorically neutral:
Make fewer presumptions about the user
Take care in what you assume about the reader of a particular topic. Make your topics work for a wide range of assumptions.

• Degree of interest
• Purpose for reading
• Prior knowledge.

Avoid stylistic whiplash
Use a transparent style. *Horton* 1994. On-line documentation/principle/nodes

221 Eliminate unnecessary material. Do not repeat what the user already knows.
Horton 1994. On-line documentation/principle/information content.

223 Cross-reference rather than include.

In paper documents, we tend to explain each new term, concept, or acronym where it is introduced. In online documents this is not only wasteful, it is dangerous. Online documents are seldom read in a predictable sequence, so there is no way to tell where the user will first encounter a piece of new information. *Horton* 1994. On-line documentation/principle/information structure/design.

226 Chunking strategies:

Size online topics to fit a screen or window.

There are definite advantages to sizing all topics to fit entirely in a single screen or window. The user does not have to scroll down to read all of the topic. The display does not waste space on a scroll bar or next-screen buttons.

This strategy works well in products for children and novices. Information kiosks and CBT lessons for beginners cannot rely on the user knowing how to scroll the display.
Horton 1994. On-line documentation/guideline/nodes.

229 Size for the purpose of the topic:

Size for the user
Size for the material
Size for the reading conditions. *Horton* 1994. On-line documentation/guideline/nodes.

230 Design topics for convenient display

Topic size also depends on the display size and format. Divide information into easily scanned chunks. Do not make users search for information in long displayed passages. *Horton* 1994. On-line documentation/guideline/nodes.

231 Introduction
The introduction is a concise graphic or brief paragraph that prepares the user for the content of the topic. It tells what the topic is about and lets the user decide whether it is relevant to the task at hand.
• Provide a smooth transition.
• Put the topic in context.
• Let the user decide whether to read the topic.
• Present the gist of the subject.
• Prepare the user to understand.
There are several ways to introduce a topic:
Graphical overview.
Elaborate the title.
Summary of the contents. *Horton* 1994. On-line documentation/guideline/node.
232 Title
Make titles context free
Make titles understandable
Make titles scannable
Make titles thematic. *Horton* 1994. On-line documentation/guideline/information design.
233 Why create links? Links -
Show relationships
Connect related facts
Cope with scatter.
Share topics. *Horton* 1994. On-line documentation/guideline/links.
234 Link triggers
Make clear what the link does
Link triggers are the words, icons, buttons, or other objects the user selects to have the link display its destination. We must make link triggers apparent to users without cluttering the screen. Users must recognize which part of the screen are active and which are not. *Horton* 1994. On-line documentation/guideline/links.
235 Make link triggers apparent
• Highlight the trigger.
• Use a special mark.
• Display triggers beside sources.
• Display on menu.
• Change cursor when over trigger. *Horton* 1994. On-line documentation/links.
236 Word triggers clearly.
Horton 1994. On-line documentation/guideline/links.
237 Allow multiple destinations. *Horton* 1994. On-line documentation/guideline/links
238 Make selecting links easy.
With mouse or pointer
By skipping to it.
With key or button. *Horton* 1994. On-line documentation/guideline/links.
239 Define specific types of link. *Horton* 1994. On-line documentation/guideline/links.
241 Before you link...
Do not include links for superficial or circumstantial associations. Include links only if the association is one the user will understand and consider important. Before adding a link, ask:
• Will a significant number of users want to stop reading here and start reading the destination of the link?
• Do users need the information at the end of this link? If the primary users are novices, reduce links to obscure uses of commands, unlikely errors, and advanced procedures.
• Will users expect a link here? Learn from users. If you notice many users following a path of links A to B to C to D, then put in a link from A to D.
• Will users understand the information they find there? Do not link to information in a different language. If you must, warn the user before displaying the new information. *Horton* 1994. On-line documentation/guideline/links.
244 Organise to meet users' needs:
Organize as a paper document.

Map general questions to specific answers. *Horton* 1994. On-line documentation/principle/information design.

245 LAYER DOCUMENTS

Progressively disclose more information

Make the first response to the point.

Provide hints before help

Show synopsis before details

Summarize first. *Horton* 1994. On-line documentation/guideline/information design.

246 ENGINEER A READING SEQUENCE

Presenting information in the wrong order hampers learning, yet in intricately organized online documents, that order is entirely in the hands of the unguided user. As a writer of online documents, you cannot anticipate all the possible ways a user can reach a given topic in an intricate document. You can, however, define a preferred reading sequence that presents information in a coherent order that ensures context.

Design topics so that they are understandable - or at least not confusing - if read out of order. At the same time, design them to encourage and support reading patterns that will help users. *Horton* 1994. On-line documentation/guideline/information structuring.

Design topics for combining. *Horton* 1994. On-line documentation/guideline/information structuring.

247 ENGINEER A READING SEQUENCE - Set up recommended reading paths. *Horton* 1994. On-line documentation/guideline/information structuring.

248 ENGINEER A READING SEQUENCE - Begin with an overview. *Horton* 1994. On-line documentation/guideline/information structuring.

250 ENGINEER A READING SEQUENCE - Build along the path. Build on information earlier in the reading trail. Add to the reader's knowledge and understanding. *Horton* 1994. On-line documentation/guideline/information structuring.

251 ENGINEER A READING SEQUENCE

Summarize at departure points. At the end of the reading trail, put a brief summary. A summary does three things for the user:

- Marks the end of the reading trail.
- Refreshes the reader's memory.
- Ensures all users see the critical ideas. *Horton* 1994. On-line documentation/guideline/information structuring.

252 STANDARDIZE COMMON TOPIC CLUSTERS - If groups of the same kinds of topics occur over and over throughout your online document, standardize how you organize them. This consistency makes navigation easier for the user. *Horton* 1994. On-line documentation/guideline/nodes.

253 ENGINEER EFFECTIVE DIALOGS - Interact with the user. *Horton* 1994. On-line documentation/principle/user interface.

254 ENGINEER EFFECTIVE DIALOGS - Do not overload the user. *Horton* 1994. On-line documentation/principle/user interface.

255 ENGINEER EFFECTIVE DIALOGS - Do not try to outsmart the user. *Horton* 1994. On-line documentation/principle/user interface.

256 ENGINEER EFFECTIVE DIALOGS - Let the user control the dialog. *Horton* 1994. On-line documentation/principle/user interface.

257 ENGINEER EFFECTIVE DIALOGS - Simplify access to the document. *Horton* 1994. On-line documentation/principle/user interface.

258 ENGINEER EFFECTIVE DIALOGS - Let users close the document at any time. *Horton* 1994. On-line documentation/principle/user interface.

259 LET USERS DO MORE THAN READ - Let users add annotations. *Horton* 1994. On-line documentation/principle/user interface.

260 LET USERS DO MORE THAN READ - Let users talk back. *Horton* 1994. On-line documentation/principle/user interface.

261 LET USERS DO MORE THAN READ - Let users copy and print. *Horton* 1994. On-line documentation/principle/user interface.

262 LET USERS DO MORE THAN READ - Let users modify documents (sometimes). *Horton* 1994. On-line documentation/principle/user interface.

263 LET USERS DO MORE THAN READ - Allow multiple authors. *Horton* 1994. On-line documentation/principle/user interface.
264 USE FAMILIAR METAPHORS:
Book metaphor
Spatial/travel metaphor.
Control panel
Photo album
Television
Games
Showroom
Magazine
Pile. *Horton* 1994. On-line documentation/principle/user interface.
265 ANSWER QUESTIONS QUICKLY
Respond rapidly.
Do not respond too quickly.
Respond consistently.
Explain delays.. *Horton* 1994. On-line documentation/principle/user interface.
266 THE "LOST IN HYPERSPACE' PROBLEM
Keep the design simple. *Horton* 1994. On-line documentation/principle/user interface.
267 THE "LOST IN HYPERSPACE' PROBLEM
Link moderately and methodically.
Use hypertext as a means, not an end.
Not too many links. *Horton* 1994. On-line documentation/principle/user interface.
268 THE "LOST IN HYPERSPACE' PROBLEM
Keep the user oriented
Display the topic name and path to it.
Number each topic and display.
Display a You are here icon.
Show a roadmap of the document.
Give users a clear starting point. *Horton* 1994. On-line documentation/guideline/user interface & navigation.
269 DECIDE WHERE DOCUMENTS APPEAR - Display documents in the most useful location. With limited space to display documents and programs, you must choose whether to display documents as complete screens, in a fixed area of the screen, or as a resizable, reshapable window. *Horton* 1994. On-line documentation/principle/user interface - screen display..
270 DECIDE WHERE DOCUMENTS APPEAR
Reveal new information predictably.
Replace the current display.
Add independent windows.
Pop up a temporary window.
Inject information into the current display.
Scroll new information onto displays. *Horton* 1994. On-line documentation/guideline/user interface - screen display..
271 DECIDE WHERE DOCUMENTS APPEAR
Manage window proliferation.
As a rule of thumb, users can handle three to five windows before they start to become confused and annoyed. If you have good reasons, exceed that limit, but take special steps not to overwhelm users.
Horton 1994. On-line documentation/guideline/user interface - screen display.
274 DO NOT PACK INFORMATION TOO TIGHTLY
Simplify the display. *Horton* 1994. On-line documentation/guideline/user interface - screen display.
275 DO NOT PACK INFORMATION TOO TIGHTLY
Replace text with graphics. *Horton* 1994. On-line documentation/guideline/user interface - screen display.
276 DO NOT PACK INFORMATION TOO TIGHTLY
Condense text for continuous reading. Higher density may actually increase reading speed for continuous text. Horton 1994. On-line documentation/guideline/user interface - screen display.
277 DO NOT PACK INFORMATION TOO TIGHTLY

Use blank space actively. *Horton* 1994. On-line documentation/guideline/user interface - screen display.
278 ARRANGE THE DISPLAY LOGICALLY
Divide the display into functional areas.
Shape functional areas to fit their content.
Put smaller functional areas around the edge.
Balance the display. *Horton* 1994. On-line documentation/guideline/user interface - screen display..
279 ARRANGE THE DISPLAY LOGICALLY
Group related items.
If you must present many separate pieces of information, present them in a few small groups. Distinguish different types of information in the display. Make sure controls for the content are distinct from controls for the system presenting the content. Also, clearly distinguish procedural from conceptual information.
To group related items:
• Cluster them together, surrounding the group with blank space.
• Draw a box around a group of related items.
• Display related areas using similar visual characteristics, such as color, font, or reverse video. *Horton* 1994. On-line documentation/guideline/user interface - screen display..
280 ARRANGE THE DISPLAY LOGICALLY - Arrange items in a familiar, logical pattern. *Horton* 1994. On-line documentation/guideline/user interface - screen display..
281 ARRANGE THE DISPLAY LOGICALLY - Anticipate and guide the user's eye movements. *Horton* 1994. On-line documentation/guideline/user interface. - screen display.
282 ARRANGE THE DISPLAY LOGICALLY - Put action areas near the centre of attention. *Horton* 1994. On-line documentation/guideline/user interface - screen display..
283 USE COLOR TO COMMUNICATE - Use color functionally. Avoid irrelevant and unnecessary color. Use color to:
• Differentiate and join
• Focus attention
• Speed up search
• Express a range of values. *Horton* 1994. On-line documentation/guideline/user interface - screen display.
284 USE COLOR TO COMMUNICATE
Select color carefully.
Pick color for a purpose.
Apply the user's color conventions
Balance colors.
Avoid garish colors.
Horton 1994. On-line documentation/guideline/user interface - screen display.
285 MAKE TEXT LEGIBLE
Use well-formed character shapes
Simple shapes.
Familiar proportions.
Large dot matrixes
Distinct strokes.
Fuzzy fonts. *Horton* 1994. On-line documentation/guideline/user interface - screen display - text.
286 MAKE TEXT LEGIBLE
Increase contrast.
Legibility depends on adequate contrast between text and background. *Horton* 1994. On-line documentation/guideline/user interface - screen display - text.
287 MAKE TEXT LEGIBLE - Avoid ALL UPPER CASE
Use full capitalization sparingly, especially in the body of text. Words in ALL CAPITAL LETTERS are harder to recognize and are read more slowly than the same words in lower-case letters. *Horton* 1994. On-line documentation/guideline/user interface - screen display - text.
288 MAKE TEXT LEGIBLE - Increase line spacing and blank space. *Horton* 1994. On-line documentation/guideline/user interface - screen display - text.
289 MAKE TEXT LEGIBLE - Keep lines short. *Horton* 1994. On-line documentation/guideline/user interface - screen display - text

290 MAKE TEXT LEGIBLE - Left justify text rather than fully justify the text. *Horton* 1994. On-line documentation/guideline/user interface - screen display - text.

291 MAKE TEXT LEGIBLE

Do not overemphasize reading text.

Emphasis text that the user scans for but does not read continuously. However, use emphasis only lightly in text the user must read and not just recognize. Set headings and instructions off from the body of text by blank space. If you must emphasise key words in the body of text, highlight only a few and use only subtle emphasis mechanisms, such as underlining, boldface, italics, or a slight difference in color. *Horton* 1994. On-line documentation/guideline/user interface - screen display - text.

292 MAKE TEXT LEGIBLE - Size type for prominence and legibility

The proper size for text depends on several factors, including the reading distance; screen resolution; contrast between text and background; visual acuity of the user; and whether the text is scanned, read word-by-word, or read character-by-character. Display text large enough so that users recognize characters, and yet small enough so that they can see entire words or phrases at a glance. Between these two extremes is a zone of optimum legibility. *Horton* 1994. On-line documentation/guideline/user interface - screen display - text.

293 EMPHASIZE SCANNING TARGETS

Dynamic techniques:

Change the display

Blink

Dynamically display text

Avoid visual distractions

Reverse video

Color

Brightness, boldness, line width

Boxes and borders

Size

Blank space. *Horton* 1994. On-line documentation/guideline/user interface - screen display.

294 USE SHORT, SIMPLE, FAMILIAR WORDS

Use concrete terms

Avoid unnecessary computer jargon

Beware secret jargon

Call things what the user named them

Horton 1994. On-line documentation/guideline/information presentation.

295 ENSURE ACCURATE READING

Take care with small important words

Take care with prefixes

Avoid overabbreviation

Use only standard, easily read symbols

Do not rely on punctuation. *Horton* 1994. On-line documentation/guideline/information presentation.

296 SPEAK SIMPLY, DIRECTLY, AND ACCURATELY

Write simple sentences -

Write active sentences

Make positive assertions

Keep sentences simple

Horton 1994. On-line documentation/guideline/information presentation.

297 EXPRESS IDEAS PRECISELY

Speak directly to the user

State quantities exactly

Horton 1994. On-line documentation/guideline/information presentation.

298 DO NOT MAKE THE COMPUTER SOUND HUMAN

Horton 1994. On-line documentation/guideline/user interface.

299 WRITE FOR THE LITERAL-MINDED USER

Horton 1994. On-line documentation/guideline/information presentation.

300 AVOID WISECRACKS *Horton* 1994. On-line documentation/guideline/information presentation.

301 APPLY A CONSISTENT STYLE THROUGHOUT

Words
Grammatical structures
Abbreviations
Spelling
Special conventions. *Horton* 1994. On-line documentation/guideline/information presentation.
302 ASSUME NON-SEQUENTIAL SCANNING
Keep paragraphs short
Emphasise the new, the novel, the surprising
Avoid the as-shown-above syndrome
Avoid blind references
Horton 1994. On-line documentation/guideline/information presentation.
303 WRITE GLOBALLY
If your document will be translated into other languages or used by people of widely different cultures, take steps to preserve meaning and avoid confusion
Write for translation
Take care with spoken words
Localize other things too. *Horton* 1994. On-line documentation/guideline/information presentation.

Appendix 1b: Procedure guidelines

33 Design Goal 2: Appropriateness of the document to the organisational sponsor and individual users' needs. *Brooks*,1993. Technical communication/principle/stakeholder/user requirements.
39 Design Goal 4: Economy of effort, technological constraints, and financial resources necessary to publish hypermedia documents. Don't let available equipment, technology, or project managers determine the hypermedia design. *Brooks* 1993. Technical communication/principle/general design strategy.
40 Hypermedia design tip 8: Don't spend too much time or money creating an on-line document; you may be wasting time on a document that may be rapidly out of date, or cluttered with special features that may complicate distribution. *Brooks* 1993. Technical communication/principle/general design strategy.
41 To achieve an artistic level of esthetically (sic) pleasing screen design, the hypermedia document designer must go beyond these or any other set of general principles. *Brooks* 1993. Technical communication/principle/interface design & aesthetics.
42 1. Place content over computer technology. The content must take prominence over the technology and this requires that designers suppress the desire to use a technique just because it exists. Every use of the technology must be in the best interest of the content. *Heller* 1991. Interactive multimedia: computing access in public spaces/guideline/general design strategy.
43 2. Plan for the broadest user profile. The designer must establish many different user profiles, including the handicapped (PL 73-112, EDUCOM, 1990). Given those profiles representing the physical, cognitive, aesthetic and life experiences of the user, a designer can formulate a composite profile from their sum. *Heller* 1991. Interactive multimedia: computing access in public spaces/guideline/user analysis.
44 3. Design for state of the art - 1[?]. The designer has to ensure that the CAPS [Computing Access in Public Space] is as modern as possible by the time of delivery without being too avante garde. If the computer technology pushes the development envelope, it might intimidate the user or overpower the content. However, if the technology appears out-dated, the message may not appear current or relevant. Hardware independent prototypes allow the designer to forestall hardware decisions. *Heller* 1991. Interactive multimedia: computing access in public spaces/guideline/analysis - general design strategy.
48 An exhaustive test phase is needed to insure that the application is robust and able to withstand constant usage. *Heller* 1991. Interactive multimedia: computing access in public spaces/guideline/interface design.
54 1. List the necessary or desirable features of the application, with an emphasis on quantity. *Waterworth, Chignell, & Zhai* 1993. Hypermedia/guideline/user analysis & specification
55 2. Group into related features, bearing in mind the hierarchical relationships amongst the features. Waterworth, *Chignell, & Zhai* 1993. Hypermedia/guideline/methodology - functional specification.
56 3. Design interface features, such as icons and controls. *Waterworth, Chignell, & Zhai* 1993 p461. Hypermedia/guideline/methodology - initial design.
57 4. Test the designs from 3, above, on users/ If necessary, return to 3 to improve/modify the designs. *Waterworth, Chignell, & Zhai* 1993 p461. Hypermedia/guideline/methodology - iterative design.
58 5. Induce micro-models from successful interface features. *Waterworth, Chignell, & Zhai* 1993. Hypermedia/guideline/methodology - middle design.
66 6. Integrate micro-models into an overall interface model. *Waterworth, Chignell, & Zhai* 1993. Hypermedia/guideline/methodology - middle design.
67 There is a large body of information organized into numerous fragments. *Shneiderman* 1989. Hypertext/principle/preliminary decisions.
68 The fragments relate to each other. *Shneiderman* 1989. Hypertext/principle/preliminary decisions.
69 The user needs only a small fraction at any one time. *Shneiderman* 1989, p115. Hypertext/principle/preliminary decisions.
70 1. Know the users and their tasks. Users are a vital source of ideas and feedback; use them throughout the development process to test your designs. Realize you are not a good judge of your own design because you know too much. Study the target population of users carefully to make certain you know how the system will really be used. Create demonstrations and prototypes early in the project; don't wait for the full technology to be ready. *Shneiderman* 1989. Hypertext/principle/user analysis.
71 2. Meaningful structure comes first: Build the project around the structuring and presentation of information, not around the technology. Develop a high concept for the body of information you are

organising. Avoid fuzzy thinking when creating the information structure. *Shneiderman* 1989. Hypertext/principle/information design.

72 3. Apply diverse skills: Make certain that the project team includes information specialists (trainers, psychologists), content specialists (users, marketers), and technologists (systems analysts, programmers), and that the team members can communicate. *Shneiderman* 1989. Hypertext/principle/project management.

75 7. Work from a master reference list: Create a master reference list as you go to ensure correct citations and prevent redundant or missing citations. Some hypertexts automatically construct this list for you. *Shneiderman* 1989. Hypertext/guideline/middle design & construction.

97 Rule 19. When adapting for hypermedia presentation documents created according to book technology, do not violate the original structure. *Landow* 1991. Educational hypertext/principle/text conversion.

98 1. Decide who your users are. *HyperCard Stack Design Guidelines*, 1989. Hypertext/guideline/user analysis.

99 2. Decide what the stack's subject matter is and what it is not. *HyperCard Stack Design Guidelines, 1989* . Hypertext/guideline/design strategy.

100 Decide the purpose of your stack. HyperCard *Stack Design Guidelines*, 1989. Hypertext/guideline/Initial design decisions.

101 Decide how much information you will cover. *HyperCard Stack Design Guidelines*, 1989. Hypertext:/ guideline/initial design decisions.

102 Decide what resources the stack will require and estimate how long it will take to build. *HyperCard Stack Design Guidelines*, 1989. Hypertext:/guideline/preliminary design decisions.

118 Write for the screen, not for paper. *HyperCard Stack Design Guidelines* 1989. Hypertext:/guideline/user interface- screen.

121 Guideline 7: Plan on changing your stack several times. *HyperCard Stack Design Guidelines* 1989. Hypertext/guideline/iterative design.

122 ...think of several solutions, design all of them to a certain point, and then choose one for further development. *HyperCard Stack Design Guidelines* 1989. Hypertext/guideline/iterative design.

123 Guideline 8: Test early, test often, and listen to your reviewers. You may be tempted to wait until your stack is "perfect" before you show it to anyone. To do so could mean disappointment and a lot of additional work. It's better to get feedback early and often by establishing regular tests or reviews of your stack-in-progress. Listen to what people tell you. Watch what people do when they try to use your stack. See what pitfalls people find and what solutions they try, without offering your comments.
Test your help function as well as the stack. Ask users whether the help system provided the information they needed, and what information they wanted but couldn't find. If your testing shows that your users need to get help often, perhaps your stack has other design flaws. Maybe the structure is too confusing or is not well suited to the subject matter. *HyperCard Stack Design Guidelines*, 1989. Hypertext/guideline/iterative design.

152 3.2. Identify all objects that make up the system.
In the hypertext framework based on the Structure of the Intellect model, both nodes and links can be considered as objects. Nodes are classified into six semantic types - detail, collection, proposition, summary, issue and observation. Links are of two major types - convergent and divergent. Convergent links can be further classified into specification, membership, association, path, alternative, and inference links. Divergent links can be categorized into elaboration, opposition, speculation, branch, lateral and extrapolation links. The emphasis of this framework is the association of semantics to nodes and links. Organizing nodes and links semantically helps manage the hypertext network and its sub-networks better. It would also help reduce ambiguity, disorientation, and cognitive overhead. *Balasubramanian & Turoff* 1994. Hypertext/guideline/interface design-information design.

153 3.3. Identify all actions/functions that can be performed on these objects. Separate them into generic actions, explicit actions, and control functions.
Actions and functions are the operations which can be performed on these 6 node types and 12 link types. Actions can be generic, explicit, or control-like. [more] *Balasubramanian & Turoff* 1994. Hypertext/guideline interface design.

154 4. Identify modifiers/filters that select subsets of objects.
The framework identifies all nodes and links by their semantic types. This type information can become a modifier or qualifier to the specific object. For example, a search can be performed on all nodes of type

"detail". Similarly, we can also request for all links of type "specification". Type information and keyword information can be combined to narrow down the search criteria. For example, there can be a query to retrieve all nodes of type "detail" containing the keyword "hypertext". Modifiers for templates can include the type of a template, the author's name, creation date, etc.. *Balasubramanian & Turoff* 1994. Hypertext/guideline/information design.

155 5. Identify strategic choices which allow the user to accomplish a specific task.
Strategic choices include user interaction with the system in order to accomplish a specific task. These can be treated as landmarks [ref,]. Strategic choices might include overview requests, (displays of summary nodes), structural and content query facilities, navigation mechanisms, editing tools, display options, audit trail mechanisms, linearization techniques, and backtracking facilities. The inherent nature of hypertext is that there is really no need for strategic choices. An node or link in the entire network can be a strategic choice. In a true hypertext system each node should be self sufficient and complete. Also, the set of strategic choices need not b e the same every time the user interacts with the system. For example, the user can directly access required nodes through query mechanisms. This feature. must be available even without traversing the network. Hence, a query facility can be considered as a strategic choice. Strategic choices can also include an overview diagram of the template (its contents and links), the ability to find out from which master template a duplicate was created, the ability to edit the master template etc. *Balasubramanian & Turoff* 1994. Hypertext/guideline/information retrieval & access design.

156 6. Identify lateral classifications - objects and actions that relate to each other.
Hypertext provides the inherent capability of creating lateral classifications. The ability to create a lateral link t another version of a node or to an annotation (a detail or observation node) can be considered as lateral classification. *Balasubramanian & Turoff* 1994. Hypertext/guideline/information design

157 3.7. Identify the formats of objects, parts of objects, menus etc..
Though this section is more applicable only when designing the actual interface for a particular application, the designer must decide on the formats for nodes and links. Formats should also be defined for overview diagrams, indexing mechanisms, query facilities, and results of a query. A template can be considered as a pre-formatted collection of webs that can be directly used by the author. *Balasubramanian & Turoff* 1994. Hypertext/guideline/interface design.

158 3.8. Identify lists of objects.
This is related to identifying explicit actions and modifiers. It is necessary to identify the kinds of lists that might be produced - lists of nodes or links or templates classified by semantic type, lists of nodes last visited, lists of nodes or links last modified etc.. Lists can also include user created annotations. *Balasubramanian & Turoff* 1994. Hypertext/guideline/early design.

159 3.9. Identify reactive choices that can be performed on a list or a set of objects.
Balasubramanian & Turoff 1994. Hypertext/guideline/early design.

160 3.10. Identify processes or functions that share information.
The ability to create objects (whether nodes or links) or list them is common across the system irrespective of the kind of object. Hence, these can be shared processes reacting differently based on the type of object being acted upon (similar to the object-oriented concept of operator. overloading). The shared process of creation can be extended to create templates. *Balasubramanian & Turoff* 1994. Hypertext/guideline/information design.

161 11. Identify all user interaction states. *Balasubramanian & Turoff* 1994. Hypertext/guideline/interface design.

162 12. Identify necessary help throughout the system. *Balasubramanian & Turoff* 1994. Hypertext/guideline on-line help design.

163 13. Identify all error conditions. *Balasubramanian & Turoff* 1994. Hypertext/guideline/error avoidance.

164 14. Identify the screen layout - workspace, control area, status area, message area etc.. *Balasubramanian & Turoff* 1994. Hypertext/guideline/screen design.

165 Early and Continual Focus on Users.
Direct contact-through interviews, observations, surveys, participative design-to understand cognitive, behavioral, attitudinal, and anthropometric characteristics of users-and their jobs.
Decide (a) who the users will be and (b) what they will be doing with the system *Gould* 1988. HCI/principle/general.

167 Integrated Design.

All aspects of usability evolve in parallel; All aspects of usability under one focus. *Gould* 1988. HCI/principle/ usability.

168 Early-and Continual-User Testing

Early on, intended users do real work with simulation and prototypes; their performance and reactions are measured qualitatively and quantitatively. *Gould* 1988. HCI/principle/iterative design.

169 Iterative Design

System (functions, user interface, help system, reading material, training approach) is modified based on results of user testing; test cycle is repeated.

The key requirements for iterative design are: Identification of required changes. An ability to make the changes. A willingness to make changes. Gould 1988. HCI/principle/iterative design.

170 Checklist for achieving Early User Testing :

— We made informal, preliminary tests of a few user scenarios-specifying exactly what the user and system messages will be- and showed them to a few prospective users. *Gould* 1988. HCI/guideline/requirements analysis.

171 Checklist for achieving Early User Testing (cont.): — We have begun writing the user manual, and it is guiding the development process. *Gould* 1988. HCI/guideline/design strategy.

173 Checklist for achieving Early User Testing (cont.): — We have used simulations to try out the functions and organisation of the user interface. *Gould* 1988. HCI/guideline/iterative design.

174 Checklist for achieving Early User Testing (cont.): — We have used mockups to try out the functions and organisation of the user interface. *Gould* 1988. HCI/guideline/iterative design.

175 Checklist for achieving Early User Testing (cont.): — We have done early demonstrations. *Gould* HCI/ Guideline/ iterative design.

176 Checklist for achieving Early User Testing (cont.): — We invited as many people as possible to comment on on-going instantiations of all usability components. *Gould* 1988. HCI/guideline/iterative design.

177 Checklist for achieving Early User Testing (cont.): — We had prospective users think aloud as they used simulations, mockups, and prototypes. *Gould* 1988. HCI/guideline/iterative design.

178 Checklist for achieving Early User Testing (cont.): — We used hallway and storefront methods. *Gould* 1988. HCI/guideline/iterative design.

179 Checklist for achieving Early User Testing (cont.): We used computer conferencing forums to get feedback on usability. *Gould* 1988. HCI/guideline/iterative design.

180 Checklist for achieving Early User Testing (cont.): We did formal prototype user testing. Gould 1988. HCI/guideline/iterative design.

181 Checklist for achieving Early User Testing (cont.): We compared our results to established behavioral target goals. *Gould* 1988. HCI/guideline/iterative design.

182 Checklist for achieving Early User Testing (cont.):We met out [sic] behavioral benchmark targets. *Gould* 1988. 1988. HCI/guideline/iterative design.

183 Checklist for achieving Early User Testing (cont.): We let motivated people try to find bugs in our system. *Gould* 1988. HCI/guideline/iterative design.

184 Checklist for achieving Early User Testing (cont.): We did field studies. *Gould* HCI/guideline/iterative design.

185 Checklist for achieving Early User Testing (cont.); We included data logging programs in our system. *Gould* 1988. HCI/guideline/iterative design.

186 Checklist for achieving Early User Testing (cont.): We did follow-up studies on people who are now using the system we made. *Gould* 1988. HCI/guideline/iterative design.

187 Checklist for carrying out Iterative Design: — All aspects of usability could be easily changed, i.e. we had good tools. *Gould* 1988. HCI/guideline/software/tools.

188 Checklist for carrying out Iterative Design (cont.): We regularly changed our system, manuals, etc., based upon testing results with prospective users. Gould 1988. HCI/guideline/iterative design.

189 Starting points. There are several beginning points, prior to talking to potential users: Define the system. Follow-on Systems. New Influential Systems. New Technologies. User Circumstances. Journals, Proceedings, Demonstrations. Other Designers and Consultants. Workshops and Short Course. Standards, Guidelines, Development Procedures. Gould 1988. HCI/guideline/feasibility.

190 Use hypertext for browsing large networks of information: In general hypertext is useful for exploring small, independent but interrelated topics in a large body of information (Shneiderman 1989). *Horton* 1990. On-line documentation/guideline/feasibility.

191 Use hypertext for browsing large networks of information (cont.):
To teach concepts. Hypertext helps self-directed learners explore a body of information at their own pace. In learning systems, hypertext lets learners reference other materials to clear up questions or to explore private interests (Thorsen & Bernstein, 1987; Wolman, 1988). *Horton* 1990. On-line documentation/guideline/feasibility.

192 Use hypertext for browsing large networks of information (cont.): For highly annotated documents. Literature, poetry, theology, politics, art and other fields of research commonly require examining annotated source works as well as reviews and commentaries. Because hypertext links comments to the source, it can display the source in its original form, bringing comments into view only as needed by the user (Kreitzberg and Shneiderman 1988). *Horton* 1990. On-line documentation/guideline/feasibility.

193 Use hypertext for browsing large networks of information (cont.): For problem-solving systems. Hypertext systems are valuable for research, note-taking, and writing when questions are ill defined and information is incomplete. They encourage exploring and brainstorming, allowing the user to build up a solution from related pieces of information. *Horton* 1990 On-line documentation/guideline/feasibility.

194 Use hypertext for browsing large networks of information (cont.): For loose collections of interrelated documents.
Hypertext is ideal for creating collections of loosely structured documents (Trigg & Irish 1987). Hypertext automates cross-references within a document and from document to document. In doing so, it makes relationships explicit and easy to explore. *Horton* 1990 On-line documentation/guideline/feasibility.

195 Use hypertext for browsing large networks of information (cont.): To model and teach organisation. Hypertext deals explicitly with organization as an element of communication. It lets the user examine the structure of a body of information, and it lets the author shape that structure directly. For these reasons it is ideal when the structure of information is especially important. *Horton* 1990. On-line documentation/guideline/feasibility.

196 Do not to [sic] use for quick simple tasks. The complexity of creating and using hypertext rule it out for simple projects and tasks that must be done quickly. Do not use hypertext for strictly reference documents, especially quick-reference documents. Having to jump from topic to topic is too time-consuming for seekers of specific reference information. *Horton* 1990. On-line documentation/guideline/feasibility.

198 Use hypertext as a spice.
Hypertext is like cayenne pepper - a little goes a long way. Endless possibilities for exploration offer endless possibilities for getting lost. . . . Do not make hypertext an end in itself but a characteristic to give an organised body of information. *Horton* 1990. O n-line documentation/guideline/general design strategy.

201 Plan for expansion.
Many of the first-written topics of hypertexts use links sparingly because the writer has not yet created topics to which to link. To get round this problem, plan the entire document in advance and list all topics or create dummy topics. Embed links as you write even though their destinations are not complete. *Horton* 1990. On-line documentation/guideline/initial design/future development.

203 Start with simple projects.
Hypertext is the most difficult form of online documentation. Creating it requires applying all the principles in this book with judgement and wisdom. To learn to build hypertexts, start with a project:

• With which you are familiar.
• That has identifiable, independent topics.
• Whose interrelationships are clear to you. *Horton* 1990. On-line documentation/guideline/general.

205 Originality.
Originality for its own sake cuts across the primary objective of helping the reader. '...you should explore the frontiers of the tools at your disposal........but put the results of such experiments into your hyperdocument only if they improve its value to the reader.' *Martin* 1990. Hypertext/principle/general design.

206 Craftsmanship in the details
The author who wants to create an excellent hyperdocument should pay attention to the details. Use interesting illustrations. Employ an artist who can make the artwork beautiful. Search for the perfect metaphor. Refine the structure repeatedly. Polish the English. Mies Van Der Rohe said of architecture "God is in the details."

CD-ROMs may be too large to have everything painstakingly polished. It may be appropriate to have a nucleus which is elegant and beautifully thought out, with hyperlinks to large amount of detail or reference material which is unpolished but well indexed. *Martin* 1990. Hypertext/principle/general design.

207 Thinking about the reader.

It is worth repeating that the first rule for any communicator is to understand her audience. The author should think carefully about who will use the hyperdocument:

- Who are the readers?
- Why are they reading the hyperdocument?
- What do they know already?
- What technical words do they understand?
- What are their problems, needs and issues?
- How can the author help them?
- What constitutes value for the reader when he uses the hyperdocument?
- How can this value be maximised?. *Martin* 1990. Hypertext/guideline/feasibility.

208 Search for visual forms of representation.

To a large extent, hyperdocuments are a visual medium. The author should develop skills with visual presentation as well as skill with words.

Good textbooks have many diagrams. A hyperdocument should be more visual in its representation than a textbook. The author should examine the subject matter and decide how he can effectively use diagrams, lists and hierarchies. The author should constantly categorize and make the categorization visual.

Hyperdocuments on most subjects should not have more than 200 lines of contiguous text (in multiple envelopes) without a bullet list, chart, diagram, or summary. *Martin* 1990. Hypertext/guidelines/information design.

124 Guideline 9: When you've finished, check the stack one last time.

This step is the exhaustive, meticulous final exercise that can make a stack gleam. No matter how much you've checked before, check the production version again. Enlist new eyes to help. Deliberately do the wrong thing and see what happens. Bring in new testers and offer a reward for every problem they find. *HyperCard Stack Design Guidelines* 1990. HyperCard/guideline/final checks.

209 Usage of a grid

The definition of a grid should be a starting point for the design of a hypertext interface. Apply the grid on all pages in an application. Try to use the same grid for other applications as far as possible. This forces you/the author to put the same objects always on the same place, which is important for readers. Otherwise, the readers are disturbed by 'flipping' buttons or icons while browsing through an application or they have to understand different layouts each time they start a new application.

Snap all objects to that grid on pixel level. Otherwise, tiny little inconsistencies are caused, which also burden an extra cognitive load on the reader. *Meusel, Eickmeyer & Koslowski* 1995. Hypertext for on-line help/guideline/user interface - screen design.

215 Teamwork

Last but not least we think that it requires a team of multiple experts with strongly varying backgrounds for designing the interface of a technical hypertext. For example, we have discussed the need of a professional designer for the general layout, for good typography, color design and so on in the above listed issues. Below, we will see that domain experts (for example: implementors of software) and technical writers should also be members of the interface development team, because of their deep understanding of the domain, and also computer scientists for efficient implementation and users for evaluating the hypertext interface. A project where a designer, several computer scientists and technical writers were involved is described in (Zeis, Meusel 1993). *Meusel, Eickmeyer & Koslowski* 1995. Hypertext for on-line help/guideline/team management.

207 Put documents online only when the value of having them online exceeds the cost. *Horton* 1994. Online documentation/guideline/feasibility.

208 Design top-down, build bottom-up. *Horton* 1994. On-line documentation/principle/strategy.

209 Learn who uses the product and why. *Horton* 1994. On-line documentation principle/feasibility.

210 consider the age of users. *Horton* 1994. On-line documentation principle/user analysis.

211 [Consider] The tasks they perform. *Horton* 1994. On-line documentation/principle/user analysis.

212 Set clear objectives. *Horton* 1994. On-line documentation principle/feasibility.

213 Observe what questions users ask. *Horton* 1994. On-line documentation/principle/user analysis.

214 Use state-of-the-art peopleware:
Assemble a multitalented team
Organize as a team
Ride out the dip in morale.
Anticipate learning costs.
Horton 1994. On-line documentation/guideline/feasibility.
215 Select software carefully:
Know what you want to do
List essential features
Rank these features
Compile a list of candidates
Gather information
Pick the best three to seven products
Prepare a chapter-sized document in each
Test the finalists with a larger document
Decide
Communicate your decision
Horton 1994. On-line documentation/guideline/feasibility.
216 Conversions are never simple, easy, or cheap. *Horton* 1994. On-line documentation/principle/feasibility - information design.
217 Design for efficient production.
Unless you have unlimited time and budget, you must take care to produce online documents efficiently. good management techniques help control costs and keep projects on schedule. *Horton* 1994. On-line documentation principle/feasibility - strategy.
218 Design for efficient production
Build from reusable components
Track tasks and components
Build from a document database approach
Distribute development
Develop for multiple platforms
Automate routine chores of development and maintenance. *Horton* 1994. On-line documentation/principle/strategy.
Complete the main idea in each topic. *Horton* 1994. On-line documentation/guideline/information design.
225 Chunking strategies:
Size online topics to fit paper pages.
• When converting one-page-per -item paper documents
• For documents that will be read from paper
• For users familiar with the layout of paper pages. *Horton* 1994. On-line documentation/guideline/information design.
227 Chunking strategies:
Let the size vary. Variable sized topics are necessary when:
• Units of information vary widely in size.
• The display size is unknown. *Horton* 1994. On-line documentation/guideline/information design.
228 Resist arbitrary limits.
• Challenge editorial standards
• Don't write to fill the hole
• Break old habits
• Remove software limits. *Horton* 1994. On-line documentation/principle/strategy
240 WAYS OF CREATING LINKS
Manual construct links
Compile links from tagged files.
Infer links from the content.
Create computed links. *Horton* 1994. On-line documentation/guideline/linking.
MAKE EACH VERSION A SEPARATE LAYER.
Horton 1994. On-line documentation/guideline/design strategy.
AVOID COMMON ORGANISATIONAL PROBLEMS:

Russian-dolls structure.

Layer information moderately. Avoid a Russian-dolls structure with endless layers of topics within topics within topics. Don't make the user burrow through many layers to get to a simple piece of information. Each additional action is another chance for the user to make an error or give up. Balance layering with short-cuts to tunnel through layers and go directly to specific facts. *Horton* 1994. On-line documentation/principle/information design.

AVOID COMMON ORGANISATIONAL PROBLEMS:

Fire-hydrant syndrome.

Many online documentation systems suffer from fire-hydrant syndrome, so named because getting a bit of information from one is like trying to take a sip of water from a fire-hydrant. They force the user through an excessively long sequence of displays. These structures often result when the writer fails to take into account the small display size of the computer. *Horton* 1994. On-line documentation/principle/information design.

249 AVOID COMMON ORGANISATIONAL PROBLEMS:

Premature organization

. . . . when ideas are classified and structures assigned before the writer knows enough to do the job well. Once recorded, such ideas are difficult to reorganize, so an ill-formed or incomplete structure perpetuates itself. *Horton* 1994. On-line documentation/principle/information design.

DESIGN SCREENS METHODICALLY

1. List items to display.
2. Organize the items.
3. Lay out the display.
4. Emphasis important items.
5. Test.
6. Revise. *Horton* 1994. On-line documentation/guideline/information design.

Appendix 2

Appendix 2(a): The Electronic Prospectus Design Process

INTRODUCTION

A hypertext electronic prospectus was designed and constructed, based on the existing College prospectus. The information content was imported, edited and restructured manually into a hypertext structure. The work on the prototype HyperCard stacks is described here. A discussion of the issues, problems and decisions involved is found in Chapter 5.

FEASIBILITY

A number of areas were explored. IT provisions in Scottish schools and colleges were investigated. The existing paper prospectus was analysed. The resources available were investigated.

This revealed that the majority of Scottish schools had a predominance of Apple computers, whilst FE colleges tended to be more PC oriented. Potential users were identified as being school and FE college students, their family and friends, careers teachers and local authority careers advisers.

Time was not in itself considered as a constraint in real, as opposed to research study, terms, although a working electronic prospectus would be subject to the same kind of time scale as the current paper prospectus, being required each year in sufficient time for next year's applicants to make their choices.

Existing prospectuses for the College were traditional in form and layout. Evidence from research into prospective students' information sources (Keen and Higgins 1990, Keen and Higgins 1992, SED 1993) suggested that the traditional prospectuses in general somewhat under-used, lending support to the idea that a hypertext version could find a niche.

Resources were, as indicated, limited. As well as limitations in terms of time and money, there was no specialist software and technical support available. This precluded the use of programmed design solutions. The design and development was in the hands of the author, who had some limited experience in hypertext design, having designed and developed an online information system providing user help with software packages. There were a number of Apple IIcx computers running HyperCard available, and a small budget for research expenses. A number of hypertext software packages were investigated for production of the electronic prospectus document, including Director, ToolBook, HyperCard and Guide. Budgetary considerations at the time ruled out Director and Guide, although they both had attractive features. In particular, Director has attractive multimedia design features, and Guide has good facilities for working with existing text files. A tentative decision was made to use HyperCard with the option of converting the product to ToolBook to provide a PC-based version, partly because of the author's previous experience with HyperCard, and partly because of the Apple bias in Scottish schools. By developing in HyperCard, the resulting document could if required be converted into ToolBook by using a utility designed for that purpose. Another alternative popular today is the creation of WWW pages. When this research originally commenced there was no history of its use, and it was not considered as an option. Using HyperCard was subject to the constraints that have been documented elsewhere (eg. Kahn 1989), but it had the advantage of familiarity.

REQUIREMENTS ANALYSIS: STAKEHOLDER ANALYSIS, USER ANALYSIS AND TASK ANALYSIS

A formal requirements specification document was not drawn up. However, a number of analyses were completed, and these are described below.

Stakeholder analysis

The stakeholders are all those having an interest in the system under construction. This means not just the users but all the other parties concerned in support, and in the outcomes of use.

An analysis of the stakeholders and some of their possible requirements was made by identifying the various possible scenarios of use. This attempted to identify the possible 'stakeholders' in the system and to discuss the scenarios within which the 'QMC electronic prospectus' might be used. It also suggested the tasks that each of the scenarios requires the system to perform. This analysis only discussed the above in general, as this stage was to precede a more specific user and task analysis. The purpose here was to identify the possible interaction scenarios with the stakeholders in an attempt to gain some insight into requirements for the electronic prospectus document.

The initial target group for the product was identified as school, and FE college pupils, with the intention of a finished version being distributed to schools and colleges to act as a tool for promoting the College. Information content would be much like that of the current prospectus. With the product being sent to schools and colleges, the teachers and other staff become stakeholders. Relatives and friends must also be considered.

Scenarios

This technique simply involved making a list of the likely interested parties, and exploring the possible ways in which they might use, or be affected by, the system. A large number of scenarios arise with each of the stakeholders identified, and these are described below. It is unlikely that all the potential scenarios are identified here.

Schools and colleges

Various scenarios arise within the school and FE college context. The likely context will be the classroom, computer workshop or other study room. The sessions with the document could be group sessions led by a teacher or lecturer, individual consultations with a teacher/lecturer or teacher, or just groups of pupils or individuals using the document alone.

This implies that there can be few assumptions made about the expertise of the users. Indeed it is safest to assume that the user will have very little expertise, so the system should therefore take account of this. It is essential that all level of users should be able to interact with the system without undue problems, that is novice users should have no trouble in operating the system and 'expert' users such as careers teachers should not be held up when looking for information. The growing number of mature students entering higher education suggests that computer experience may be relatively low, and this has to be allowed for.

The types of scenarios anticipated would require the document to allow the following.

- General browsing of the information to allow an overview of the material and identification of specific areas of interest.
- Some way of retracing the material already covered
- Possibility for group use means the information must be presented clearly enough to allow a group of users to gather round a monitor and comprehend the information.
- The number of novice and inexperienced users anticipated means that a clear information structure is required to facilitate navigation.
- Good support for navigation is required for the same reasons
- And also good help and early instructions for use

More specific tasks for the system become apparent when an individual user of the system is considered. An individual user could require specific information from the system. The document should therefore:

- Allow the user to retrieve specific information answering or related to their query.
- Identify material not covered by the document, and identify supporting sources - contact addresses, references, etc..

The document may have a significant PR role outside of attracting students. Therefore it is important that an implemented document look s good, and professional, works efficiently and holds the users' interests.

Queen Margaret College

Most likely interaction seen as being on an individual basis. Context of use in offices, workshops. The user would be either a member of the academic staff, administrative staff, checking and updating information, or a student, perhaps looking for information for course change.

Users in this context are more likely to be computer-literate but not necessarily to have encountered hypertext. Conventional information retrieval methods are likely to be more familiar, and speed of access could be more important. So:

- Provision for novices should not interfere with quick retrieval of information.
- Multimedia features like music sequences, movie clips should be easily turned off to enhance quick traversal and retrieval and to avoid potential disturbance in office environments.

Open days and other public demonstrations require ease of use, reliability and professional stability and appearance

Overseas students.

Another expected use for the system would be to distribute it to potential students who are geographically remote from Glasgow and are unable to visit. In particular, overseas students are identified as potential stakeholders in the system. This means that the system should:

- Allow tasks to be carried out by users whose first language may not be English.
- Ensure understanding is not dependent on a knowledge of UK culture, society etc..
- Be reliable and capable of use without support from the College

Other stakeholders

The IT Centre staff: could be involved in technical matters in authoring and user support.

College administrative staff - could be involved in dealing with complaints over faulty product - not normally an issue with paper prospectuses

Use of contemporary technology and the generality of the information available in the document would also mean that general visitors to the College could be shown the system.

User analysis

This section first examines in more depth the nature of some of the principle users in designing an electronic prospectus, and then goes on to look at the user's task. In each case existing knowledge is examined, prospective ways of improving that knowledge are explored, and a description of what was done for this project ensues.

Existing knowledge

Where the user population is relatively significant, there is a good chance that there will be existing information about that population. This is certainly the case with prospective students. Some research into the audience for the Queen Margaret college prospectus had already been conducted by the College's public relations department. This consisted of focus groups of school students. This material had been used to redesign the current prospectus (the edition for 1994).

<u>Investigations into the nature of the UK student population</u>

As noted in the literature review, some work on the potential student population has been done. Two key surveys on the English population were conducted by Keen and Higgins (1990, 1992). The Scottish population was studied by the SED (1993) using a telephone survey method.

Keen and Higgins (1990) first investigated what the so called 'traditional' sixth-form applicants to higher education knew about higher education in general. They used a survey approach to elicit participants' knowledge of a wide range of topics, including knowledge of the different types of higher education institution, the grants available, the courses, validation, potential students' sources of knowledge about higher education and the recognition of individual institutions. The survey was based on 2000 schools in England and Wales. Scotland and Northern Ireland were not included as differences in educational systems would have necessitated the use of a different questionnaire. The schools were tested in four batches of 550 in a series of phases in order to assess sixth formers at various stages in their progress towards higher education application.

This was followed by the same team in 1992, with a similar method being employed to examine mature students on access courses, mature applicants, their parents and their friends.

The other major source is the HEIST survey of students at college (Roberts and Keen 1992) in which 5650 students at 100 higher education institutions who had just completed the first year of their studies were surveyed. This examination of their views and experiences examined four aspects of the student experience: careers and educational guidance in schools, choosing institutions, the transition from sixth form/college into higher education, and the student as consumer. The first three are of particular relevance.

Non-student users

Non-student users are also important. These include parents and friends, teachers and careers advisors. Parents and friends have been examined in the work by Keen and Higgins (1990, 1992) cited above, whilst teachers and careers advisors appear to be neglected.

So what is known?

What was clear when these surveys were conducted is that there was a substantive difference between the two main groups of higher education applicants, the school-leavers and the mature applicants. This manifests itself in significant differences in knowledge about higher education and in priority of information need. The latter is considered below as an aspect of the user's task.

As far as the former is concerned, a number of interesting points emerge. In the case of applicants at school, it appears that there is an increase in knowledge of higher education with age up until the end of the first term in the upper sixth form, with a decline in knowledge after that. It appears that applicants are sensitised by the process of application, with schools providing education and materials at this stage, and then, perhaps because of exam pressure or because the decision-making process is over, lose interest. So one segment of the potential users is a fairly compact age group. Another point is that knowledge of the universities amongst young applicants proved greater than that of the polytechnics and Colleges of Higher Education (CHEs), with the latter particularly disadvantaged. The recent promotion of the polytechnics may have changed this: the use of the term 'university' makes their status clear whilst at the same time giving new and unrecognised names. In this context it is important for CHEs like QMC to ensure that they are distinguished from Colleges of Further Education. In terms of information content, this means a consistent emphasis on university-level status, both at the level of corporate identity, and at the level of information content for courses and departments.

As far as mature student applicants are concerned, it seems that their knowledge of higher education is less than might be expected, less than that of their counterparts in the upper sixth. Keen and Higgins

(1992) suggest that this reflects their lack of involvement in a higher education directed culture, and the consequent limited access to information and advice, and relative isolation from other applicants. The general tendency was for male prospective students to be more knowledgeable on most points than females. This implies that general supporting information about the ethos, culture and aims of higher education be a part of any prospectus. In the sixth form group, there were some differences in knowledge between comprehensive, public school, Sixth form college and FE college students. It is hard to say what practical effect this is likely to have at the prospectus level, except to say that such heterogeneity of user population requires that more information rather than less should be provided.

Examining non-student users, Keen and Higgins' (1992) survey showed parents to be more knowledgeable than mature applicants, and often more knowledgeable than sixth form applicants. As Keen and Higgins point out, this level of knowledge is a good indicator of their level of interest in higher education, and this level of interest is not reflected in the way in which higher education was publicised at the time of the report. Examination of typical material reveals this is still largely the case.

Conclusions from the user analysis

The most significant points to emerge were the existence of quite substantial differences in knowledge between sixth form applicants and mature applicants. Another market segment not considered by any of this work, but one which recent trends in recruitment suggests is significant is the overseas applicants. The information needs of these groups are likely to be different. Some smaller differences appear between the sexes, and between applicants from different educational sectors. It is important for the hypertext designer to recognise first that the user population is heterogeneous, with the implication that choice in the pattern of use and access to information is likely to be important. This suggested a structure where information was provided on a need-to-know basis, rather like Shneiderman et al.'s (1991) 'details on demand' or Carroll et al.' s (1988) 'minimal manuals'. This would enable the more knowledgeable to find relevant information with the minimum of effort, whilst others could search for necessary detail. Such an approach does not by itself deal with users whose awareness and understanding is insufficient to allow them to search effectively.

Task analysis

Task analysis consisted of two distinct phases. Firstly it was useful from the author's point of view to develop a conceptual model of the user's task. McKerlie and Preece suggest a broad categorising of task goals for hypertext use into four types:

- finding an answer to a particular question (ie searching task)
- gaining a sense of scope for the information (ie browsing task),
- exploring a particular concept (eg learning), and
- collecting and tailoring information (eg. organising or synthesising)

(McKerlie and Preece 1993 p36)

According to this framework, prospectus use has aspects of all these tasks, in that users may be trying to answer a number of specific questions (such as 'what course?', 'what is the college like', 'what are the entry requirements?'), they may be browsing (what courses can I do that involve caring?'), they may be learning ('so that's what dietetics is'), and they may be collecting and tailoring information (for instance collecting lists of course requirements for comparison).

Looking more closely at the information seeking aspect, Marchionini (1992) describes the information-seeking task as a form of problem solving in which 'either or both the information sought (problem) and the search process (solution path) may be simple or complex.' This can be described in terms of five functions that form the information seeking process: defining the problem, articulating the problem, selecting the source, extracting the information and examining the results.

These functions exist together in a non-linear relationship where any of the different functions may be engaged at different points in the search process. So for instance examination of results is not necessarily an end to the information seeking process but may lead to a redefinition of the problem or an extraction of new information.

This is followed in many cases of prospectus use by a decision-making task, whether this be to select a specific college and course, or something less defined such as a decision not to go into higher education, or a decision as to a general area of study. This cannot be easily unravelled from the information-seeking task as the process involves the user constantly checking the information found against the decision-making task for relevance and usefulness.

Apart from the career professionals, the majority of users are likely to be relative novices, although they are likely to gather expertise in prospectus use a) through repeated concentrated use during the course selection period, and b) to a lesser extent, through following a series of courses at successively higher levels.

The task analysis

The original prototype involved a more or less direct transformation of the existing text prospectus. This formed the nucleus of the information to be included as it represents the institution's evolved view of what the information content should be. It was apparent from discussion with members of the Department and representatives from the PR department that no great deviation from this would be expected.

Empirical task analysis at this stage consisted of a series of pattern notes conducted with a range of participants including prospective students and new students. New students were chosen partly for their easy availability but also for their recent familiarity with the course choice process: they provide recent experience and the benefits of hindsight.

The pattern notes

Pattern noting, a task analysis technique suggested by Jonassen et al. (1989) for task analysis in hypertext was used:

Procedure

A range of students, including first year and prospective postgraduate students were asked to draw representations of the knowledge that they required to choose a course and a place of study, and the resulting networks were analysed. The students were not selected for representativeness but rather for their potential for generating a full range of required information. Twelve informants participated. The first year students were chosen on the basis that having recently been through the experience of choosing a college and course, they had the experience still fresh in their minds, plus the benefits of hindsight. This approach reflects the logic used with such qualitative methods as focus groups, (Morgan 1988). The procedure is repeated until the majority of items and connections being noted become redundant.

A complete list of the factors included in the pattern notes was then compiled. This was separated into information elements and personal considerations, and edited as necessary. The information elements were then connected diagrammatically according to connections made by the informants. This was initially done by hand, owing to the difficulties of performing such a task on a small screen. All the elements were written out, and the connections noted in the pattern notes added. They were then rearranged using a drawing package that enabled links to be preserved whilst the elements were moved to produce a clearer structure (D02, D03, this appendix).

INFORMATION CONTENT ANALYSIS

Materials

The existing prospectus was examined carefully, and a table of headings and sub-headings constructed (Table P03, this appendix). Most of this material was already broken into chunks in the form of short paragraphs. Availability of existing prospectus text files and graphics was explored. There were no problems in obtaining prospectus files, but these were in PageMaker form, and required conversion back into ASCII text files.

INFORMATION DESIGN

The first information design decision was whether to retain the existing document structure or to develop a new one on the basis of the new knowledge gathered. In any case where there already exists a paper version of the hyperdocument proposed there are the two possibilities here - to convert the original text or to author from scratch to suit the hypertext medium. At this point an early prototype was constructed in order to assess the position. This was based on an old HyperCard shell from the Hypertext '89 Conference which was adapted. This had been developed in order to show that HyperCard really could be used to produce true hypertext. The advantage was that it already contained routines for making an index, map, and contents, and in particular for enabling the use of 'hot' words in the text that were actually linked to the word rather than superimposed over it[1].

The original text files from the prospectus were used, along with scans of the photos. The text was not marked-up in any easily accessible way, being in the form of PageMaker files. Utilities are available which can be used to convert hierarchical text structures into linkable structures, but were not available in this instance. This meant a laborious process of manual conversion, from PageMaker files to ASCCI files, and then importing into HyperCard text fields. This was time-consuming, and it proved

---

[1]This was before the addition of this feature to the standard version of HyperCard

necessary to keep a careful record of the files and their translated versions. A consistent labelling strategy proved vital.

The next task was to arrange it into appropriate chunks. This proved difficult. At first the text files were imported with their existing structure. The question was, how and where to organise the chunking. The choices were to chunk whilst the material was in ASCII format, or to import it into HyperCard and then chunk. Chunking whilst the material is in ASCII format had the advantage that material could be saved under separate and relevant file names, then imported into a single card field using a HyperCard import script. This only worked with whole text files, which meant that a large number of text files had to be created, one for each chunk of information. This meant even more care with file names to avoid confusion. Alternatively, the tactic chosen, existing relatively large text files could be imported into a temporary 'text files' stack , and then either chunked whilst in that stack, or cut and paste in chunks from that stack into the actual prospectus stack. Again, this was a tedious process, complicated by the strain that chunking imposes in cognitive terms: the author has to make decisions about the information content whilst engaged in the mechanical and repetitive task of importing and copying across text.

HyperCard is capable of a variety of hypertext structures, but they must normally all be authored by hand. The underlying structure of HyperCared is linear, or more accurately, circular or looped. That is to say, that if no links are added by the author, it is only possible to progress from card to card by moving from one to the next in line, until one is returned to the start. The net result of these two phenomena is that it is easiest to produce a linear structure in HyperCard. By importing the files in sequence, the linear structure of the printed prospectus was replicated. Some efforts were made to add extra links, but the linear structure proved to inhibit this. The result was really nothing more than an electronic page turner with an active contents list and a few extra links as shortcuts to certain areas of the information.

Hence it was decided to author from scratch, using the pattern note analysis as a basis for the new structure. A table-diagram was constructed, listing all the pattern note elements, and these were linked (D01, this appendix), according to links in the pattern notes. It was apparent from this that a number of elements had not been considered by the participants, and that to proceed solely on the basis of the pattern notes would lead to a rather quirky form of prospectus. This was not a major problem in it had always been considered necessary to include elements from the printed prospectus. It was also evident that the elements included were not linked as one might expect: certain logically apparent connections were not made by the users, and some elements were totally unconnected. It was decided to modify the structured diagram to include obvious logical connections, for instance 'history of city' and 'character of city'. The material was re-organised at this point to clarify the link structure (D03).

This gave a somewhat unstructured network, so the information was then supplemented by developing a hierarchical structure for the information. A structured list of the elements was produced, and this was merged with topics extracted from the printed prospectus (P03) to produce a new table, P04 (this appendix). From this a hierarchical structure was developed initially on a bottom-up basis, with individual elements being sorted into categories until a top level of a few large categories was reached. This was converted into diagrammatic form. Links were then added across the hierarchy as indicated by the cumulative pattern note to give a cross-referenced hierarchy (D04, this appendix). This formed the basis of the information design.

DESIGN SPECIFICATION

A formal design specification was not produced for the first version. For the second version, the structure diagram provided the main specification. This only provided details of links between nodes and node clusters.

INFORMATION CONTENT: CREATION AND INPUT

For the main authored-from-scratch version, the material already imported into HyperCard was retained, and a linking structure imposed using the structure diagram for guidance. This saved re-importing the material. The shell was modified to give a short linear opening sequence to a main menu. This functioned as the first level in a flat hierarchy. Two kinds of card could then be created, using a scripted utility, part of the original shell. This gave head of section cards and content cards with a sub-heading. The sections corresponded with the second level in the hierarchy. Clicking the mouse on the contents item could allow a jump either to the head of a section or to subsections within it. Some of the text was edited and re-chunked to fit modified headings and sub-headings. Editing was necessary because HyperCard's frame or card-based approach meant a limit to the amount of text which could be displayed on screen. Then interface decisions about for instance font sizes also have a direct impact on the amount of text on a card. This is complicated in HyperCard's case by the possibility of using scrolling fields on each card, which enable cards to hold rather more than their size would otherwise dictate. The lowest categories, ie those shown at the bottom of the hierarchy diagram, (D04,this appendix) were distinguished typographically with in subsections. The course description sections took a standard structure as described in table P05 (this appendix).

LINKING AND ACCESS

Here the existing HyperCard shell made some of the linking decisions automatic. A script in HyperCard's accompanying scripting language, HyperTalk looked up the titles of cards and entered them into an active contents listing, giving section and card titles which could be clicked on to jump to those items. It was then a relatively simple matter to use background buttons showing on most cards to allow a linear route through most sub-sections, that is, the lowest level of the hierarchy. There were no extra access mechanisms at this stage.

USER INTERFACE DESIGN

A number of conventional principles and guidelines for user interface design for hypertext exist, and an extensive body of knowledge. This material was examined in some depth for the taxonomy of design guidance. However, this research took place concurrently with the development of the electronic prospectus, so the final results were not available. Sets of interface design principles and guidelines for hypertext were already available however, and two of these, by Shneiderman (1989) and Hardman and Sharratt (1990) were used as a basis for a set of principles and guidelines as follows:

- Meaningful structure comes first
- Consistency- the task should be structured so that a consistent presentation is used and consistent sequences of actions are required in similar situations.
- Mental processing- the hypertext should not complicate the reader's information gathering tasks and impose excessive mental processing requirements. The author should be trying to (i) minimise the overall mental load by reducing the requirement for readers to remember the objects, actions, codes and abbreviations they are working with' (ii) minimise the task-specific mental processing by arranging for efficient completion of typical reader tasks.
- The document should display a suitable balance between ease of learning and ease of use. Ease of learning focuses on enabling the reader to become proficient with the hypertext with minimal training and practice. Ease of use is achieved by minimising the steps or actions taken by knowledgeable readers.
- Flexibility- the hypertext should be capable of adaptation to the needs of the user.
- Positions of the links on the screen should be obvious.
- There should be a large pointing area for option selection. Where there are small linked items in a hypertext the active area for the link can be larger than the visual representation.
- Link types should be clearly identified.
- Critical information should be highlighted. Certain parts of the information contained in the hypertext may be particularly important. This information can be highlighted but needs to be a different style from those used to denote the links in the hypertext.
- Assign visual codes carefully:

(i) Flashing should be used with great care since it can be effective the first time round but becomes irritating with frequent use.

(ii) Colour should be used as a redundant feature (i.e. colour is paired with another form of coding), an preferably with options to allow readers to select the colours they want for each code.

(iii) When using graphical representations extra care is required to produce clear codings of link types and critical information.

- Display necessary information. The items a reader requires on screen are:

(i) a clear title;

(ii) the information the reader is interested in;

(iii) indications of where the links are in this material;

(iv) other links to known places in the hypertext;

(v) sufficient context information to inform readers where they currently are.

Information should be arranged to make relationships clear. Where possible, different windows (or areas of the screen) should be used for different types of information.

Ordering of lists should be designed to assist readers' tasks, and in hypertext this can be enhanced by having multiple orderings of the same information. As well as this, what would traditionally be a linear list can be broken down into a hierarchical structure containing shorter sub-lists.

- Require low cognitive load. Readers should not be required to remember where they are when they arrive at a new screen. after actioning a link, there should be sufficient information to reorient themselves. Information on what the current section is, and where in the current section the reader is should be either immediately available on the screen, or directly accessible by a mechanism made known to the reader.

• Where immediately relevant information takes up more than one screen the reader should be able to move easily between the relevant displays.

• Terminology and wording:

(i) terminology should be familiar (or there should be easy access to definitions) and consistent;

(ii) abbreviations should be explained;

(iii) sentence structure should be simple;

(iv) instructions should be affirmative and in the active voice.

• Use graphics effectively. Icons should be clear and legible and standardised throughout the hypertext.

• Avoid reader errors in link use by clearly identifying link starting points.

• Help should always be available and obtainable through a standard procedure, e.g. a help icon is always displayed on the screen

• Minimise cursor movement. Arrange items to minimise the distance the reader has to travel to reach the links.

• Know the users and their tasks

• Respect chunking

• Show inter-relationships

• Be consistent - with formats, including font types, styles, button positions, in icon use, in creating document names

• Work from a master reference list

• Ensure simplicity in traversal

The user interface design was kept very simple in this version. A standard format was adopted for all cards containing primary information. The exceptions were menus and sub-menus, title, help, glossary and map cards. An underlying grid dividing the screen into thirds was used. A control area was reserved at the bottom of the screen. All titles and sub-titles appeared in standard fonts and positions.

TESTING AND EVALUATION

No systematic evaluation was carried out. User testing, although important, is time-consuming and difficult to arrange, and was not considered to be a central part of this research, having been the subject of extensive research by a number of workers over the years. A heuristic evaluation was carried out using the above guidelines.

Appendix 2b: Tables

This section of the Appendix contains Tables P01-4 , raw and sorted listings based on the pattern notes which were carried out as part of the electronic prospectus development described in the previous appendix.

Table P01: Pattern note data

This is an unsorted list of the data from the pattern notes - all the topics or concerns which were raised by the twelve informants. The list has been compiled directly from the pattern notes. Repeated elements have not been included, but some closely related elements have been retained rather than merged together.

geographical factors of area
geographical suitability for
personal interests
climate
landscape
kind of place college is situated in
lifestyles available
culture
social life
entertainment
types of people in area
class factors
styles
fitting in
type of academic institution - size, age, prestige
quality of teaching staff - academic, personal
reputation of teaching staff
research reputation
general reputation of institution
teaching ratios
teaching skills
teaching facilities
library facilities
IT facilities
Lab facilities
space available
political/ethical interests and values
intellectual interests
existing skills
existing knowledge
existing financial situation
existing personal/domestic situation - debts, savings
what level of study to choose?
how to fund studies
where to study
part-time v full- time study
distance learning v traditional attendance
length of course
subject to study
staying at home v moving away
accommodation costs
relative cost of living in different areas
finding accommodation
studying alongside/with career
career progression
degree level
degree kind
personal funding
student loans
grants
combination of funding methods
age
small or large institution
difficulty in funding
working to supplement grant
vocational courses
placement possibilities
career development loans - availability
grant applications
grant applications and parents - parental contributions and support
age and grants
how to fund studies - general
mixed funding methods
studying whilst working
financial considerations
crèche facilities
dependants
other commitments
personal satisfaction
personal interests
exam results
what to study
career potential
vocational courses

male-female ratio
social life
societies
size of place
places previously visited
knowing people in place
security
nearness to home
new places
where to study
parental pressure
cost of travel to place of study
access courses
extra costs associated with cities
foreign exchange possibilities
places not previously visited - new environments
academic v vocational courses
long-term (life) goals
short-term goals
pleasure/fun
improving poor existing position - ruts, dole
aspirations
current lifestyle
future lifestyle

P02: Personal factors from the pattern notes

This table lists a number of personal concerns that were elicited from the informants during the pattern-noting process.

age and experience
aspirations
current lifestyle
dependants
desire for new places
desired future lifestyle
existing financial position
existing knowledge and experience
existing skills
improving one's position
independence from parents
intellectual interests
Knowing people in college
knowing people in town
level of personal commitment
long-term (life?) goals
nearness to home
official pressure - schools, careers officers etc.
other existing commitments
parental pressure
parental relationships - as affecting funding
part-time job v course workload
part-time v full-time study
peer pressure
personal interests
personal learning style
personal satisfaction
places not previously visited - new environments
places previously visited
political and ethical interests and values
previous knowledge of university/college environment
relationship with parents
security?
short-term goals
studying whilst working
trade-off with existing position - present earnings, prospects, job satisfaction - and with other future alternatives
where are my friends going?

P03: Prospectus topics and sub-topics

This list represents the 1995 prospectus headings and sub-headings

Life at QM

About the College

Edinburgh

Communications

Culture

History

Central Facilities

Library

IT

Accommodation

Students' Association

Offices

Officers

Facilities, clubs and societies

Events

Representation

Sports facilities

Educational requirements

Age

Equal opportunities

exceptional entry

Access courses

Entry with advanced standing

Assessment of prior experiential learning

Students with special needs

Health issues

Qualifications for entry into year 2

Scottish Wider Access Programme

Application Process

What you need to apply

When to apply

How to apply - procedure and fees

How to apply - non-UCAS applications

Offers

Post-graduate and post-registration course

Principal's Forward

Fees

International opportunities

International students

Support services

Admission

International qualifications

Open Days/Careers Conventions

Financial support for international students

Staff

Departmental descriptions

Dietetics

Speech and Language Sciences

Management and Social Sciences

Podiatry

Applied Consumer Studies

Communication and Information Studies

Health and Nursing

Physiotherapy

Hospitality and Tourism

Drama

Careers advice

Counselling services

Course descriptions for the above

Appendix 3

Appendix 3a: Protocol for a single-case study

This appendix gives the full text of the protocol for the single-case study. This was designed as a guide for the Glasgow case study, as a way of controlling the case study research and thereby maximising reliability.

Introduction

The protocol is a technique suggested by Yin (1994) as a way of improving reliability in case study research and this protocol is based on methods and examples given by Yin.

1. OVERVIEW

Type of study

This study is at present a single case study. It covers a number of cases in an attempt to distinguish common characteristics. The general approach and the methods used are qualitative. It is hoped that generalisations may be made, but it is accepted that individual uniqueness and complexity may preclude such generalisation. Stake (1994) suggests that the individual researcher is faced with a strategic choice as to the depth of complexity that he or she engages with, and this directly affects the degree to which generalisation is possible. This issue is investigated more thoroughly in the analysis section.

Purpose: relation to other elements of the PhD.

The case study is intended to provide data that have a better claim to generalisability than the prototype electronic prospectus, and to examine evidence of the use of principles, guidelines and other design advice of the kind examined for the taxonomy of design guidance. These data should provide complementary evidence for the construction of an electronic prospectus design methodology. The case study does not only contribute to the overall PhD programme. It is informed by the other elements, in that data from these elements have helped shape both the case study questions in general and the specific questions which must be asked in order to answer the larger case study questions. In particular, the interview schedule is based on an early version of the design methodology that has been prepared on the basis of work on the prototype and the taxonomy. The analysis is also to be conducted on the basis of knowledge obtained in the other parts of the PhD research.

Case study Issues

Rationale for site selection

At present the number of sites that are producing or have produced electronic prospectuses are limited, although expanding. This means that issues of availability and access have a high priority. Notwithstanding this, the ideal site has the following attributes:

Recent or imminent completion of project: There is no doubt that the limits of memory and the later interpretation of events could have a distorting effect. In the case of the latter, it is hard to say whether this has a positive or negative effect on the outcome of the study, as reflection may well have provided the participants with new insights into their experience of the design process. However, it is to be expected that this reflection will over time incorporate more and more material extraneous to the case study, which whilst potentially valuable is of unknown origin.

The subject of the project should be an electronic prospectus as opposed to a campus information system. Although the two products share many features, the end users are very different. in the event of it proving impossible to identify a suitable electronic prospectus, it may be necessary to include campus information systems, but it is felt that the latter not only have different end-users, but are also developed on a much more incremental basis than electronic prospectuses which are based around complete yearly editions.

The project should ideally be commissioned by the relevant (probably information or marketing) departments within the HE institutions concerned. Prospectuses developed for research purposes or student projects, whilst interesting and often very similar in terms of software and methods used, are different in that there is no concerned client involved, and therefore the pressures and responsibilities of the development team are different. It is also the case that the standards of success are somewhat different; in the case of a research project it may be important to have some aspect of novelty or originality; in the case of student projects, the final marker is inevitably the views of the most significant member(s) of the stakeholders.

Propositions/hypotheses under examination

This research is initially descriptive, so no hypothesis is proposed. It attempts to describe what happened in some selected instances. It is hoped that it will be possible to make some proposition or propositions about an appropriate way to develop electronic prospectuses by means of developing a chain of evidence from the primary data.

Definitions

This work begins at the individual case level, and it is therefore necessary in the first instance to establish the boundaries of the individual case.

Unit of analysis

The unit of analysis can be characterised, according to Yin, in a number of ways, for instance as an individual, group or organisation. In this instance it was at first assumed that the design team would be the unit of analysis. However, as the main focus of this case study is the design process, it is preferable to take this as the unit of analysis. This makes it easier to include such characters and issues as the commissioning body and the role of other stakeholders. A further argument for considering the design process as the key unit is the role of context. The kind of project that this case study deals with is rooted in a specific organisational context. Although the project and the context are capable of definition as separate systems for the purpose of case study research, the approach taken is that the events considered are situational, and that therefore the organisational context cannot be ignored. By defining the unit of analysis as the whole of the development process, this enables the context to be considered as an integral part.

The unit of analysis is also said to present problems with defining the beginning and end of the research (Yin 1994). The subject in question could well present this kind of case, as projects of this nature may often emerge out of earlier work.

Definition of the Development Process

In this context, the development process is seen as all activities involved with the creation and distribution of the product, including the design and implementation phases. Use of the word process does not imply commitment to a linear model.

Definition of the Development Team

This means all members actively concerned with design, development and implementation of the electronic prospectus in question. It does not normally include commissioning bodies and individuals unless they are sin some way directly involved in design, development or implementation.

Relevant literature

It is normal to include relevant literature in a case study protocol in the event that others than the author are involved in data collection. This is unlikely in this case, but as the protocol could potentially be adopted by other later researchers, the sources cited above are included here:

Stake, R. E. (1994). Case studies. In: N. K. Denzin, & Y. S. Lincoln, (Eds), *Handbook of Qualitative Research* (pp. 236-247). Thousand Oaks: Sage. A useful short summary of case study issues and approaches

Stake, R. E. (1995). *The Art of Case Study Research.* Thousand Oaks: Sage. A more extensive treatment of the above, with examples and practical tips.

Yin, R. K. (1994). *Case study research: Design and methods* (2nd ed.). Thousand Oaks: Sage. Main source for this research.

2. FIELD PROCEDURES

Access

Contact in the first instance via Phil Gray, Computer Science Dept, Glasgow University, Lilybank Gardens, Hillhead, Glasgow.

Schedule of data collecting activities

1. Acquisition of example electronic prospectus. This is necessary as a first step in order to gain some understanding of the context of the development under study.
2. The hardest part to arrange is the interviews, so this should probably be next.
3. Gathering of documents, paper-based and electronic.

Resources

Access to Macintosh computers for viewing prototypes and final versions, copying files; floppy disks; stationery; suitable space for interviews. Tape recorder and suitable microphone; tapes.

Provision for unexpected events

Contingency plan for non-arrival of interviewees - back-up communications arrangement. Back-up microphone. Spare tapes.

3. CASE STUDY QUESTIONS

These are the research questions for which answers are required, which is not the same thing as the literal questions which might be asked of the participants. This can be defined in terms of Yin's classification of levels, where Level 1 is questions asked of specific interviewees, through to level 5, normative questions about policy recommendations and conclusions, beyond the narrow scope of the study.

Level 5 question (questions about policy recommendations):

What conclusions and recommendations can be offered to future electronic prospectus developers?

Level 4 (questions asked of an entire study):

What is the best way to develop a hypertext electronic prospectus?

Level 3 (questions asked of findings across multiple cases):

Not applicable at present.

Level 2 (individual case study level questions) :

Level 2 (individual case study level questions) :

This level is most problematic, in that a number of alternative questions could be asked here. The following questions were decided upon:

1. What design guidance and methods were applied by system (deve any)?
2. Which of these were useful, and to what degree?
3. What was the design process?
4. How were design guidance and methods applied in practice?
5. What design problems were experienced, and how were they resolved?

Level 1: (questions asked of specific interviewees [sources]):

The table below shows the Level 1 questions, and in the form in which they are operationalised. The latter is often the same as the former, but in some cases the research (Level 1) question has to be reformulated to make it more accessible to the interviewee.

Code	Research Questions (Level 1 questions)	Sub-question	Operational questions
	Methodology		
GM1	Was a design methodology used?		Did you use a [information systems] design methodology?
GM2	If so, which one?		If so, was it a recognised methodology, and which one?
GM3	[Elicit details of methodology from appropriate sources if obtainable, authorship, features. Otherwise obtain details from interviewees according to the questions right & below	GM3a GM3b	Who developed this methodology? Could you give me details of its main features?
GM4	What was the overall philosophy of the methodology?		What was the overall philosophy of the methodology?
GM5	What type of model did the methodology use (e.g. analytic/mathematical, iconic)		What type of model did the methodology use (e.g. analytic/mathematical, iconic)
GM6	What techniques and tools were employed? [e.g. entity modelling, data flow diagrams, project management tools, analysts' workbenches etc.]		Please give details of any techniques and tools used as part of the methodology

GM7	What was the scope of the methodology?	GM7a GM7b	What stages of development did the methodology cover? How much detail was involved at each stage?
GM8	How was development broken down into stages?	GM8a GM8b GM8c	Was system development broken down into separate stages? How was this breakdown into stages done? Was process strictly sequential, staged?
GM9	What tasks were performed at each stage?		What was the time spent on the various stages you've distinguished?
GM10	What outputs were produced at each stage? (Document, prototypes etc.)		What was the end result of each of these stages, for instance a document, working prototype?
GM11	When were the various tasks performed?		What was the overall time scale? - who did what when?
GM12	Who participates in the methodology		Who actually administers this methodology [e.g. analysts? users? both?]
GM13	To what extent was this methodology followed?	GM13a GM13b	Did you follow the methodology thoroughly? Did you experience any difficulty applying the methodology? - please give details
GM14	How was the project managed overall?		Who was in charge of the project? How were decisions taken?
GM15	what formal documentation methods were used		What kinds of documentation did you use?
GM16	To what extent were methodologies, principles, guidelines etc. adhered to?	GM16a GM16b GM16c GM16d GM16e	Did you find it easy to stick to the methodology? How much skill did it take to use? Did you find it helpful? Did you draw on principles and guidelines from other sources? If so, why was this necessary?
GM17	If no methodology was used, what alternative sources of guidance were used?	GM16a GM16b	If you didn't use a methodology, how did you structure your work? Did you use any other forms of guidance, principles, rules, guidelines? If so, which ones

	1. Analysis		
	General context		
GC1	What were the circumstances surrounding decision to build a system? I.e., whose idea, who wanted a CD-ROM prospectus, was there a genuine requirement	GC1a GC1b GC1c	Whose idea was the CD-ROM prospectus? Did the marketing/PR department show an interest? Did academic departments?
GC2	What budget was available?		What budget did you have available? What resources?
GC3	what time constraints existed?		How much time did you have?
GC4	what expertise was available?	GC4a GC4b	What expertise was available to you within the time What outside expertise could you draw on
GC5	Strategy - how was the problem defined		
GC6	At what point was the decision to use hypermedia taken?		Were you starting from scratch?
GC7		GC7a GC7b	When did you decide to use a hypermedia approach? Where did that idea come from?
GC8	What analysis of stakeholders was made?		What analysis of the various stakeholders [explain if necessary] was made?
GC9	What analysis of users?		What user analysis did you conduct
GC10	What task analysis method was used?	GC10a GC10b	Did you do any task analysis? If so, what kind?
GC11	What analysis of the information content was made?		Did you make any analysis of the potential information content at this stage, for instance an examination of the existing prospectus?
GC12	how were key concepts identified?		How did you move from the information gathered about users and other stakeholders to deciding what the system should include?
GC13	How were aims and objectives of system/system requirements evolved from this?		Did you produce a specification of requirements at this point?
GC14	Did requirement include non-functional/usability targets?		What did the specification of requirements include? [Obtain copy]

	2. Design and development		
	Timetable and project management		
DD1	Was there a team leader?		Did you have a team leader? (If so) Who was this? How was she/he chosen?
DD2	Is having a leader of any importance? If so , what>		(If no team leader) Do you think a team leader have been helpful? (if team leader) Was having a team leader helpful?
DD3	Were any formal project management methods used?		Did you use any formal project management techniques? (If so) what were these?
DD4	What was the timetable of development?		Could you fill out this table to indicate as accurately as possible what tasks and events occurred when? [See appendix]
	Resources and techniques		
DR1	How was the design team chosen?		How was the design team chosen?
DR2	How were resources decided upon, chosen & collected?		How were resources decided upon, chosen & collected?
DR3	Hardware?		What factors influenced the choice of hardware?
DR4	Software?		What factors influenced the choice of software?
DR5	What was the medium of distribution and why was it chosen		Why did you choose CD-ROM as your medium of distribution?
DR6	What authoring/programming tools were used?		Please give details of the authoring or programming software used in the project

DR7	What ancillary tools (editors etc.) were used? [If prototyping was used] what prototyping software was used?		Please give details of all ancillary tools used in the project [name should do in most cases]: *Prompt if necessary from list below* video editors: sound editors: text editors: chunkers/file splitters graphics software: prototyping software any others? [If prototyping was used] what prototyping software was used?
DR8	Design resources?		What design resources were available to you?
	Design decisions		
DD1	Where was the subject matter/information content derived from?		How did you obtain the information content?
DD2	How were design ideas generated?		How did you obtain your design ideas - [e.g.. brainstorming, experience of similar software]
DD3	Was prototyping used?		Did you use prototyping
DD4	What kind?	DD4a DD4b	What form did this take? How many iterations?
DD5	How was the information structured?		[See product]
DD6	Why was the information structured in this way?		Why did you choose the structure that you used [summarise?]
DD7	What design tools were used? (storyboards etc.)?		What techniques did you use to work out your design?
DD8	What user interface design principles were used?	DD8a DD8b	Did you follow any user interface design principles or guidelines? Which ones?
DD9	How useful were these ?		How useful were these ?
DD10	Navigation	DD10a DD10b	How much attention did you pay to navigational problems? What efforts did you make to overcome these problems? [evidence of system]

DD11	Designing nodes & links?	DD11a DD11b	How did you decide how much information to put on each screen? What influenced the choice of button types?
DD12	How were access methods decided (apart from hypertext links)?		What influenced your choice of information access methods?
DD13	Design of non-textual components - was this integral, or added on?		Was the design of non-textual elements an integral part of the design, or was it added on afterwards?
DD14	What influenced the choice between authoring and programming?		What considerations made you choose authoring/programming [as appropriate]
DD15	Evaluation with users - what methods were used and when?	DD15a DD15b DD15c	Who did you test the system on? What method(s) did you use? When did you evaluate?
DD16	What evaluation criteria were used? [Were they based on the specification of requirements]	DD16	What evaluation criteria were used? [Were they based on the specification of requirements]
DD17	What authoring strategy was chosen - start from scratch or new structure?	DD17	Did you decide to design the system from scratch, or did you adopt structures found in existing documents or systems?
DD18	At what point was a detailed system design produced? (if at all)	DD18a DD18b	Did you produce a detailed system design? When?
DD19	What form did this take? [Scripts, storyboards, flowcharts or other structure diagrams]	DD19	What form did this take?
	Construction		
C1	What was the sequence used in building the final version? [Detailed account at level of preparation of text, graphics, video etc., construction of screens, backgrounds etc.]	C1	What was the order of construction in building the final version?
C2	How efficient was the construction sequence?	C2	With hindsight, do you think this was the best way?
C3	How difficult was the timing of time dependent elements?	C3	How hard did you find the timing of time dependent elements?

C4	What system documentation, apart from designs and specifications, was made?	C4	How did you document the system?
C5	When was documentation for users prepared?	C5	When did you prepare the documentation to accompany the product?
C6	How was the system debugged [incl. procedures for link checking]	C6a C6b	How did you debug the system? How did you check the links all worked?
C7	What arrangements were made for future editions?		What arrangements were made for future editions?
	Success of the project		
SP1	How successful or otherwise were the methods employed?	SP1	Would you do things the same way again?
SP2	What were the problem areas?	SP2	What do you think were the main problem areas?
SP3	were the development team satisfied with the development process?	SP3	Were you happy with the development process?
	System acceptability		
SA1	Was the final system acceptable - to clients/commissioning authorities		Did the clients like the system? [ask team, client as well if possible]
SA2	Was the final system acceptable to users?		Did the users find the final system acceptable [my evaluation - with users + heuristic evaluation]
	Success of methodology		
SM1	Was the team satisfied with the methodology used (if applicable)		
SM2	Was the development team satisfied with any principles used other than the methodology (if applicable)		
SM3	Was the development team satisfied with the guidelines used (if applicable)		
SM4	Was the development team satisfied with the model used (if applicable)		

SM5	Did the process (methodology) used produce the best possible system?		
SM6	Did the methodology fit in with the team's way of working?		
SM7	Did the method use the available resources efficiently		

Answering the questions

Answering these questions is based on building a chain of evidence from level 1 up to level 5. At the lowest level exists the sources of evidence: these must be identified and examined and the resulting data analysed as appropriate

Sources of evidence

It is expected that there will be three main sources of evidence: software, documentation (texts and diagrams) and the individuals involved. Software includes both final and prototype versions. The former is an essential piece of evidence, as it is integral to answering question 2 above on the success or otherwise of the development process.

Methodology

It is easy to assume that because one is doing case studies that this constitutes a methodology in itself. This is incorrect (Stake 1995). A case study may involve a number of methods, both qualitative and quantitative. This study restricts itself to the use of qualitative methods, including individual and group interviews, software evaluation and textual analysis.

Interviews

The shape of these has to be to some extent dictated by the context of the individual case study, for instance varying availability of team members means that the use of group interviews may not always be possible. As indicated above, the interview schedule is fairly highly structured, although stories and digressions are not discouraged.

Software evaluation

The final software will be summatively evaluated using two methods. A heuristic evaluation will be conducted, using a schedule of heuristics based on existing guidelines plus specific elements based on this researcher's own task analysis. This will be accompanied by a user evaluation involving representatives of all potential user groups.

Textual analysis

All texts will be entered into a table giving date of document, contents and comments. This will be used as the initial basis of a chronology of the design process.

'table shell' for project timetable[1]

	Order	Date started	Date completed	People involved
Contextual study				
User analysis				
Task analysis				
Requirements specification				
Design decisions				
Prototype construction				
Prototype evaluation				
Final version construction				
Documentation				
Please include details of other events you regard as significant				

4. Analysis plan & guide for the case study report

Practical aspects

The first tasks are to transcribe any information from interviewees, and examine, collate and index all other materials. Transcription will be for content analysis. This material will be coded according to the codes included in the interview question schedule. Documentation will be numbered, and an index applied, based on a simple thesaurus. This will based on the main subject areas used in coding the interview schedule and examination of the documents materials. Other terms may be added as appropriate. The level of detail is as yet uncertain; this will depend to some extent on the complexity of the material. As it is necessary for material to be accessible for other users who may only examine the material infrequently, it is intended that the thesaurus be as accessible as possible.

At this stage it is uncertain what qualitative analysis software will be used to support the document analysis process. Options to be evaluated include NUD.IST, Word and HyperCard.

The final software product will be evaluated to assess its success. A successful method implies that the software achieves some reasonable degree of success - see below for methods of evaluation.

In general the pattern of analysis of the non-interview material will depend on the kinds of material available. The material will initially be sorted according to basic type: system content files will be separated from documentation, comments, feasibility studies, specifications, messages etc. Authorship should be identified whenever possible. Material will be dated and organised chronologically wherever possible. It will then be coded according to type and function.

[1]This approach was dropped in favour of a simple chronology, Table GT04, as seen in Appendix 3d.

Outline of individual case-study report

The case study report would most likely contain the following elements:

Introduction

Description of project

Software produced, personnel, organisational context.

Description of conduct of the study

- interviews conducted, including date, time, circumstances

To be attached as an appendix

Level 2 question analysis

1 *What methodology(ies), guidelines, principles etc. were applied by the developers of the system?*

In so far as these are overt, named and known to the developers, analysis is simply a question of listing or charting the various elements applied. A suitable format for this is a table - see below. At the higher methodological level, this may not be possible. Analysis here is perhaps to be accomplished by producing a diagram of the various tasks and the time at which these were undertaken, and then comparing the result with known methodologies in order to locate the methodology used in their context.

Firstly, the responses to specific questions in interviews relevant to this question will be codified according to a simple framework used in my original taxonomy, based on Schneiderman. Texts will be examined for evidence of use of these, and conclusions based on this evidence included in the table.

Problems: there is likely to be more difficulty where synthetic or generalised approaches are indicated, in which the developers have conflated a number of sources of varying degrees of formality, including undocumented (verbal) sources. Difficulty can also be anticipated in questions of philosophy (integral to the full description necessary for question 1), where there is a question of interpretation of surface elements at a deeper level.

This will be followed by an interpretative account of the use of the various forms of advice.

2 *How successful or otherwise was the use of these methodologies etc.?*

Evaluation is likely to be useful in answering Question 2, as it provides evidence as to whether or not the development methods used were successful in producing an acceptable system. This may not be practical given time constraints though. If possible, the evaluation evidence would be used in conjunction with evidence that may be obtained from users and clients to provide a rating of success. It may not be possible to provide a straightforward scale for assessing success, as success in these contexts may be dependent on perspective, and be subject to trade-offs.

3 *How was the methodology implemented in practice?*

Initial analysis will consist of a history of the project, based on all the evidence and referencing all evidence. This entails some effort at arranging documentary material in chronological order. How easy this will be depends on the form in which documents are presented. Text documents may be dated. Documents in electronic form may have a 'last modified' or 'created' date. This may be reliable, but some form of correlation is necessary. The obvious way to do this is to make sure that some dates are obtained from the interviewees, hence the inclusion of the dates table shell, but this again may be unreliable - for instance the interviewees may have prepared for the interview by referring back to documentation.

Chronology

The datestable may be adequate here.

Structure diagram

This may be useful

Level 3 question analysis - Cross-case analysis

Not applicable at present.

Level 4 question analysis.

Based on the Level 2 material, and referring to that material as evidence.

Level 5 question analysis.

This analysis refers in turn to the Level 4 analysisto give final conclusions and recommendations. This gives a chain of evidence right from the raw data through to the final conclusions.

Validation

The study should ideally be validated by asking participants to comment on the project history and by asking other researchers to check the material.

Notes for (group) interviews

Time

Expected time 1-2 hours. If anyone has to leave early, that's fine

Ground rules

Three things:

One person speaks at a time

Side conversations make for distractions and difficulty in transcription.

I'd like everyone to participate

Confidentiality

Obviously this material is for a PhD that will eventually be in the public domain. Your names won't appear in any written document. However, as I'm engaged in case study research, it may be possible for others to identify participants. I shall endeavour to make sure that this can't happen.

I'm recording the discussion. If anyone wants to say anything 'off the record, they can let me know, and I'll switch the machine off.

Purpose and approach

This is a case study intended to find out what happens in the hypermedia design and construction process. What I have is a number of questions which I'd like responses to, but I want you to feel free to tell me any stories or anecdotes that relate to the questions under discussion.

[Switch on tape]

[Group meetings] Could you all please introduce yourselves first, so I can identify your voices later.

Appendix 3b: Table GT05 - analysis of case study evidence: Glasgow Visual Open Day electronic prospectus

This constitutes a link in the chain of evidence from the raw data to the higher level questions asked of the case study. It is a compilation of the evidence from the various data sources in the form of answers to the Level 1 questions which were asked, with references to the various parts of the data which provide the relevant evidence.

Explanation of transcript table

The transcript table shows from left to right: the research question, a code giving its relationship to the Level 2 questions of the case study, a question code for reference to the Level 1 questions, the intended operational questions - that is the interpretation of the research questions for the interview, and a full transcript of the interview including the actual questions asked by the interviewer. On the far right is a summary answer to the Level 1 question, and a column for miscellaneous notes and comments. The Level 1 answer forms the basis of answers to the Level 2 questions, which are reiterated below. Answers to the Level 2 questions are found in Table GT06 (Appendix 3e).

The Level 2 questions:

1. What design guidance and methods were applied by the developers of the system (if any)?
2. Which of these were useful, and to what degree?
3. What was the design process?
4. How were design guidance and methods applied in practice?
5. What design problems were experienced, and how were they resolved?

Research questions (Level 1 questions)	*Level 2 question type*	*Code*	*Operational questions - original schedule*	*Interview transcript*	*Summary answer*	*Notes and comments*
1. General questions						
Methodology						
Was a design methodology used?	1	GM1	Did you use a design methodology ?	*Did you actually use any kind of formal design methodology?* We didn't use any formal design methodology, not at all	No	This seems surprising given their background
If so, which one?	1	GM2	Which one was that? (if applicable)	n/a	–	
[Elicit details of methodology from appropriate sources if obtainable. Otherwise obtain details from interviewees as to design procedure according to the questions right & below	1	GM3	How did you organise the design?	*In that case, do you want to fill me in on [inaudible]* Well, the way it started was, I'm a 4th year now, but it was a third year project, it was a team project to develop a prototype for a new information system for the Department, so at that stage we weren't looking to produce a final product it was more sort of getting us to develop a prototype and ideas and come up with design decisions and work as a team. So, in the prototyping stage, the first thing we had to do was, to look at the old and existing, which was then a HyperCard stack which was done some four or five years ago *Was that the one Jill Russell did?* Yes it is, still , still around, so we played about with that, we couldn't really - we didn't really think it was going to be so out of date, and we're sitting playing with that we just, five minutes, five minutes playing with that and we just didn't want to use it at all, it was boring, it was black and white, it was. . . and I would've that though by that stage they'd've replaced it but they hadn't, so we got a few ideas about that, about the sort of information that was going be in it, then after that we went on to look at what was currently available in the way of multimedia design tools so we looked at a number of different things, there was Director, I think SuperCard or [inaudible] or something, was follow up to HyperCard *Yeah* *and . .*	Started with review and analysis of earlier version of a prospectus in HyperCard (G002) Next stage, structure maps and storyboards (G088)	There is a more up to date version - 1993 I think. Maybe they didn't see this, or maybe it just looked out of date... SuperCard of course wasn't a follow-up.
				That was the one with colour wasn't it? Then looking at the different forms of media was applications of QuickTime movies now, sound, full colour pictures, the lot. Mmmm, next stage was developing kind of a storyboard, drawing out maps of how the system was going to be organised, and from then on we just - it was a number of us working on it - we sort of took a section each and started to develop it and come up with the ideas of our own, trying to build a basic framework first and then trying organise how we would put it all in , into the...		

What was the overall philosophy of the methodology?	1	GM4		*If I sort of ask you some more specific questions, I've got a question here, what was the overall philosophy of the methodology? You might not be able to answer that.* What do you mean by that *Mmm I think in terms of mm where you're coming from, a strictly systems background or er..* It was, it was more like, like, it was one of the more un - I wouldn't say unstructured, but it was more like you could do what you wanted to, and it was all just, I wouldn't say it was more of - it's not really a scientific-type development - it's more sort of a yeah, having your own ideas and doing what you think's best rather than following rules and regulations, so it's rather more enjoyable in that aspect *Right*	–	Not very well handled. Not really an answer of the kind required.
What type of model(s) did the methodology use	1	GM5	What type of model(s) did the methodology use (e.g. analytic, mathematical, iconic)	n/a	–	
What techniques and tools were employed? [e.g. entity modelling, data flow diagrams, project management tools, analysts' workbenches etc.]	1, 4	GM6	Did you use any specific techniques or tools?	*Did you use any more specific sort of techniques and tools drawn from software engineering or anything like that in any particular respect [Interviewee answers over last part of this sentence]* You mean requirements analysis . . . yeah we were forced to do that as part as the project, mm had to do like, we did a, for the prototype we did analysis stages, requirements stages, we did do design documents but be quite honest a lot of them came near the end, it was, er, five of us, it wasn't like could say this is what we were going to do and then develop it, it was, it was more complicated than that, it was like developed and then changed, and [unintelligible] it wasn't linear, it was, you had to work backward and forward through it, mm there was, was a feasibility study we had to carry out *So you did actually, you, you did have* We have- *have a stage- by stage sort of model*	A requirements analysis; feasibility study. G005, G004, G016, G003	tendency to prompt
				yeah we were guided through that way, that way but I wouldn't say that *you didn't necessarily -* progress was made like that *Figures, yeah. Yeah, so, so what, what actually, so you had like a requirements stage which was rather loosely observed, and then you went onto actually, what did you do, did you do some prototyping?* Well, the requirements came from analysis of the existing software, and we had like a fake customer type person, they were sort of giving us ideas *Who was that?* Well, there were a few people in the department,, I don't know if you know Catherine Lyons, information officer, she does a lot of work for the..., and Phil Gray.		

				- They were the stakeholders? Yeah, well they laid down requirements for what we were going to do, and then from then on it was trying to put these requirements actually into practice and er we started, basically tried to start making frameworks for interfaces and making sure we were meeting what was in the requirements I mean obviously we weren't going to go away and so something totally different, we were going to try to adhere to the requirements, but I think the requirements were fairly loose anyway, being an information system, the only thing, the only major requirement was the target hardware platform, which was these LC2s, so we had to, and this, those requirements came in, they didn't really affect the design of the system, but sort of the quality of it, I would say, which was tailored more to the other end of the project, making sure that QuickTime movies and such could be run on the LC2s *Yeah right, so, who was actually in charge?* The prototyping stage in the third year was Richard Cooper who's in this dept here, he's er the project supervisor. I think he proposed the project, so he was in charge at that stage, and er after the presentation we gave at the end of third year, there was a number of people there, Phil Gray was one of them, Catherine Lyons, and they were actually interested in actually furthering this thing and going on and designing it. So after that stage we all basically got time to work on it over the summer and right through 'til the end of December, when everything was finally finished, so. So I think I've got the prototype, which I could show you later on, if you want to see that, and you'll see it's far different from the CD we've got, a lot different. *I was going to ask you what you used to prototype, I've got lots of questions about that sort of thing.*	Next stage, making frameworks for interfaces, referring back to requirements. Emphasis on hardware requirements	
What was the scope of the methodology?	1	GM7	What stages of development did the methodology cover? How much detail was involved at each stage?	n/a		
How was development broken down into stages?	1	GM8	Was the system development broken down into separate stages? How was this breakdown into stages done? Was process strictly sequential, staged?	n/a		
What tasks were performed at each stage?	1	GM9	What was the time spent on the various stages you've distinguished?	n/a		

What outputs were produced at each stage? (Document, prototypes etc.)	1	GM10	What was the end result of each of these stages, for instance a document, working prototype?	n/a		
When were the various tasks performed?	1	GM11	What was the overall time scale? - who did what when?	n/a		
Who participates in the methodology (e.g.	1	GM12	Who actually administers this methodology [e.g. analysts? users? both?]	n/a		
To what extent was this methodology followed?	1	GM13	Did you follow the methodology thoroughly? Did you experience any difficulty applying the methodology? - please give details	n/a		
How was the project managed overall?	1	GM14	Who was in charge of the project? How were decisions taken?		Richard Cooper (GM6)	

What formal documentation methods were used	4	GM15	What sorts of formal documentation did you use?	*What sorts of formal documentation did you use? You mentioned storyboards, maps of the system, requirements specifications, that sort* There was a problem definition, a project introduction, a user analysis of who were the possible stakehodlers, you've got teachers showing it to students, you've got interested parents, wanting to see where their kids are wanting to go, you've got the students themselves wanting to find out about Glasgow, so we had that kind of document, we had the analysis of the existing system, which was the Jill Russell HyperCard stack, and requirements document, there was a - that was a full HCI document of the requirements, the design implementation ideas, functional and non-functional requirements. I could probably print you one of them as well if you want one. *Yeah. Anything.*	Problem definition user analysis of stakeholders analysis of existing system requirements document - functional and non-functional (HCI) requirements Ideas for implementation	confusion between users, a sub-set of stakeholders, and stakeholders no an analysis of the existing paper prospectus, which some information systems design methods would expect
To what extent were methodologies, principles, guidelines etc. adhered to?	1	GM16	Did you find it easy to stick to the methodology? How much skill did it take to use? Did you find it helpful? Did you draw on principles and guidelines from other sources? If so, why was this necessary?	n/a n/a n/a		
If no methodology was used, what alternative sources of guidance were used?	1	GM17	Did you use any other forms of guidance, principles, rules, guidelines? If so, which ones	*If you didn't use a methodology, how did you structure your work?* Well, when we first started we weren't really too sure what, like how we were going to go about building this, so we managed to get in touch and see other systems that had been developed. There was one that the department had on a CD-ROM, and I think it was, don't know, it was like an ITN or BBC News CD-ROM, and it was information about all the news events that had happened over the year, and it had movies and pictures and stories in it so we got a few ideas for actual interface , you know, doing the actual colour interfaces and the layout, the different techniques of hierarchies and ways of displaying information, so basically we got a lot of ideas from looking around, from what else had been done.	Got ideas from other products eg ITN CD-ROM	

1. Analysis						
General context						
What were the circumstances surrounding decision to build a system? I.e., whose idea, who wanted a CD-ROM prospectus, was there a genuine requirement	3	GC1	Where did the idea for designing a CD-ROM come from? Did the marketing/ PR department show an interest? Did academic departments?	*Whose idea was this CD-ROM prospectus?* Well, not too sure - the actual, doing the CD-ROM prospectus, Richard Cooper was the project supervisor in third year, and he proposed it, but as a prototype, so he must have initially had that idea of, maybe we could do one, but we'll need to see how feasible it is, and how much work's really involved in it, so I guess it was him, but we went on actually developing it, he wasn't a part of it, he stepped down, maybe you could ask Phil Gray about it 'cos he was obviously interested in actually going forward with it. *Yeah, I was wondering where in sort of organisational terms where the motivating force was coming from.* I think from the advertising in the department, see, like Catherine Lyons and, there's a group in the department, I can't remember their name, who usually throw out lots of flyers and coloured booklets and try and interest students who come to Glasgow, but they're trying to take a different approach, and try and use CD-ROMS, so I think it was, I wouldn't say it was a marketing type thing but it was, it was obviously there to bring in the new students *They were quite keen on the idea then* The Department? *Yeah* Yes. They wanted to be the first to do it as well because well nobody's done it before. *There was some people down at the University of Glamorgan who've done one, just in the last year or so.* It's still really quiet. I can't say I've heard of other universities, OK there might be one or two here or there. I don't suppose that many people know that Glasgow's done a CD-ROM. *No that's it. The first I heard about it was a Scottish News item about it.* I only saw that a couple of weeks ago as well.	Richard Cooper? Yes Computing Science Department	Maybe . . . Think he means INTERFACE; who they had to report to in this early phase (G043)

What budget was available?	5	GC2	What budget did you have available? What resources?	*What budget did you have available?* In the prototype stage I guess it was student slave labour or something, it was just we were doing a project and they were using us to channel out ideas and stuff. There wasn't really a budget - hardware and software was there already for us to. . . Available to get resources quite easily from other departments, like video cameras from the Media department; we had the University photographer taking photographs for us, so there was a lot of collaboration there, a lot of help. *That's another one of my questions there, what expertise was available to you at the time? So you had a photographer?* That was one of Catherine Lyon's ideas, because she knows him, he's done a few pictures before for the booklets and such - we managed to get about twenty or thirty photographs from him. We shot the videos ourselves, in a room just through there, with a video camera, and so it's a bit, the design of the CD-ROM, and the prototyping, it was a whole lot of experience for us, and we had to sort of write scripts for people, bring people in, this was friends and such and film people, run about parks and things, trying to decide what people would want to see from the West End, video cameras in the rain, and er, all sorts of things. Find trying to choose backing music, as well, that was a very [indistinguishable] task. *That's kind of difficult because you can't really use anything.* But like I don't really think there was any expert help. Phil Gray's obviously got a lot of ideas about how systems should be designed and put forward, but he wasn't actually doing really sort of did any expert work for us, we sort of more or less did it ourself, except for those few photos and things.	No formally allocated budget. Departmental resources - labs, software etc.	
What time constraints existed?	5	GC3	How much time did you have?		–	
What expertise was available?	5	GC4	What outside expertise could you draw on	*What expertise was available to you within that time*	Photography. General design advice from Phil Gray (GC2 above)	
Strategy - how was the problem defined	3	GC5	How did you define the problem?		Requirements defined on the basis of analysis of existing software and input from Catherine Lyons and Phil Gray.	

At what point was the decision to use hypertext/hypermedia taken?	3	GC6	When did you decide to use a hypertext or hypermedia approach?	*Were you starting from scratch? You'd seen that HyperCard stack.* Well, we wanted to make it as impressive as we could and we wanted to show off and make it really spectacular, so from the offset it wasn't, when we started the project, it wasn't well the information system's got to have movies, pictures, sound; it's got [?] and flash, I mean they were just wanting an information system. There's er, I think there's another project going on in the department, for, doing it for Law Department, they want something similar, and there's a group doing that, but they'll not, they've just asked to develop an information system, they're concentrating more on just providing a text thing, rather than while we've got multimedia, but I think, being a project we wanted to make it quite fun and quite exciting to develop, so we weren't wanting to sit and hack in lots of text, we wanted to take this opportunity to play with the new tools and the new toys and such, so it was getting, I think we more or less expected that we were going to put movies in; it was all basically a prototyping phase to see if it was going to be feasible for the LC2s, so it was - that's when we knew for sure what it was going to be like.	N/a - see below (GC7)	This doesn't really ask the question correctly.
When was hypertext/hypermedia adopted	3	GC7	Where did that idea come from?	–	Appears to have been passed on from Richard Cooper's original conception (GM6)	
What analysis of stakeholders was made?	3	GC8	What analysis of the various stakeholders [explain if necessary] was made?	–	Not entirely clear - see GM6.	

What analysis of users?	3/4	GC9	What user analysis did you conduct	*What user analysis did you conduct* Well, we wrote letters to a lot of schools in Scotland, primary schools and high schools. We first wrote one, just a general sort of school and department thing, to find out the statistics of what hardware and software the schools had, because at that stage we didn't know that people had LC2s, what size of hard drives they had, did people have CD-ROMs and such things. So we ended up writing letters to about twenty or thirty different schools in Scotland to try and get an idea of what we were looking for, other than just saying this is what we've got, because we wanted to try and make it as usable as possible. User analysis - basically just thinking of who the different users were going to be, as I've said, who would be interested in it: teachers, and interested students. *Did you have time to actually grab any prospective students and interview them?* No, I don't think so. there was one or two, there was one or two kids we had in when we were developing it, I think it was Phil's children, or Catherine Lyons', somebody's kids were coming in to play with it, just to see what they were, to see if they were interested in it or not, whether they felt it was going to be useful. I know myself, when I came to Glasgow Uni. I didn't know anything about Glasgow University, I just got told there was a university there, you can go there and study, you - at school it wasn't sort of said, 'go to university, this is what it offers, this what you can do, its - you've got to find out for yourselves, and we didn't really have the opportunity. I wouldn't exactly say there was a prospectus lying about our school. *These things are surprisingly hard to get hold of, and the other thing is when you do get hold of them, they're incredibly dry - the kind of detail that's in them tells you noting about the life of the place or the feeling of the place at all.* Well, they're trying to do that, have their section at the start, but the fact it's text, and it's a big book,- *- and there's one or two obviously posed pictures of students in labs and things like that.* Unless you're really interested in doing it, I don't suppose you're really going to bother reading through. It's like let's hope with this sort of thing, people who are maybe not thinking about it would actually interact with the system - 'Oh, wait a minute, that looks good.'	Contacts with schools. also education departments. Reflection.	Again a preoccupation with hardware and software, rather than the actual needs of users. Assumption that only teachers and prospective students would be using the product. Not the same thing - more a prototyping approach.

What task analysis method was used?	3/4	GC10	Did you use any task analysis methods? If so, what kind?	*You did some analysis of the actual information content, from looking at the earlier prospectus. How did you actually identify all the key elements and concepts* It's changed quite a lot over the development - from the prototyping stage anyway. Looking at the old HyperCard stack, there was a general section on the Glasgow, on Glasgow University in general; there was sort of the main section was the Computer Science department, 'cos it was built more for this department rather than the University as such. So we had that, we had those two, I think we had those two first of all, basically just through going through the different sections of that HyperCard stack and noting down the different menu structures, and the topics which were of interest, because a lot of the stuff which was in the HyperCard system wasn't really structured at all. We found it really quite confusing to get our bearing ourselves. There was a lot of stuff, there was big sections on accommodation and stuff, and how much it costs to rent a room, and stuff like that; there was like the amount of detail in one section wasn't really balanced with the amount of detail in other sections. I guess it was decided from the offset it was going to concentrate on this department, and we'd have a section on the Uni, and the breakdown on this department changed quite a lot. Initially we just thought right, what would people want to know about the department. Courses would be the first thing that people would want to know about, lab equipment maybe. Just a few idea like that, and we had, we did that in the prototype, it wasn't really, it was a few ideas , but it was only really what we had thought of that people would be interested in, and Phil Gray came up with the idea of it was going to be like, a corporate, thing, Virtual Open day, and so we were going to try an model it like having an open day so we sat down and we talked about what an open day consisted, of. You have all these different rooms where you have people you can go in and see, so we decided we'll have a section where you can go in, it's called 'Wander about'- you can go in and wander about, you can meet a few students and a few lecturers, you could go and see a course advisor, find out what you needed to get into the course, and such things, and you could have a look around the Department. So basically we were just trying to do all the things that you would do when you came on an open day: the things that you would actually want to go and find out rather than what we thought, I mean I never went to an open day *That's the advantage of a hypermedia approach, that you can give people that choice.*	_	Didn't directly ask about task analysis here. Trying too hard to be subtle.
What analysis of the information content was made?	3/4	GC11	What analysis of the information content did you do? Did you make any analysis of the potential information content at this stage, for instance an examination of the existing prospectus?		See GC10. Analysis based on the existing HyperCard stack - emphasised the structure of the information without apparently looking closely at the content, although one can't really be done without the other.	

how were key concepts identified?	3/4	GC12	How did you identify the key concepts?		See GC10. Assessed the relative balance of the various elements in the existing HyperCard stack . Weighed this against their existing ideas about the material. Integrated this with the idea of a Virtual Open Day that Phil Gray came up with, that is, they adopted information content to fit in with this structure.	
How were aims and objectives of system/system requirements evolved?	3/4	GC13	How did you evolve the aims and objectives of the system? Did you produce a specification of requirements at this point?		Worked through to a requirements statement as described above (GM6) No evidence in the interview. Formal meeting to establish requirements 18.1.94 (G042). The requirements specification appears to have been produced as a formal document after work on the prototype had begun, 24.2.94 (G042).	
Did requirement include non-functional/usability targets?	3/4	GC14	What did the requirements specification include? (if applicable)	What did the specification of requirements include? [Obtain copy]	Document D005 shows an requirements analysis in terms of 'who, what, where, when, how' basis. This is followed by a formal requirements statement listing tasks the system is to perform; a system specification in terms of features, including some general principles and guidelines; functional and no-functional requirements, hardware characteristics, error handling, quality issues etc.	Apparently written up in June, although the log G042 refers to an early version.

2. Design and developme nt						
Resources and techniques						
How was the design team chosen?	3	DR1	How was the design team chosen?	–	No evidence.	
How were the information and resources to be included chosen & collected?	3	DR2	How did you decide what material should be included? Where did you get the material?		See above - GC10, GC11.	
Hardware selection?	3	DR3	What factors influenced the choice of hardware?		See GC9 - indicates that machines available to end-users in Scottish schools were the deciding factor. (See G031 - letter to Scottish education departments.)	
Software?	3	DR4	What factors influenced the choice of software?	*What factors influenced the choice of software? You used Director in the end.* That was, when we first started the project, we had to decide how we were going to do it. I think everybody thought we were just going to use HyperCard again, or this new SuperCard, which had colour in it. We managed to find out that the Department had a few other multimedia authoring tools, and we had a look at them, and there was a few Director sort of demonstration movies, and we thought it was just so far above HyperCard and anything else that was there, and we'd heard a few names of a couple of others, but we didn't have them in the Department, so we used Director, and at that stage we were using Director 3, but when we actually got onto designing, the department had got Director 4.	Dissatisfaction with HyperCard, based on examination of existing prospectus stack, dissatisfaction with its lack of colour (at the time) and poor video handling capacity, the discovery that Director had far more potential on the basis of demonstration products ('movies') with the software. (DR4, G029 - the latter gives a retrospective account of the options considered and the reasons for opting for Director).	

				How did you actually find using Director . It's quite a steep learning curve if you haven't learnt it initially. At first it was very hard, we were not too sure what was going on. There's a few on-line tutorials and things. We had all the manuals as well. Basically just started just trying to put together a few things, and there was four of us working on it independently as well, so I mean everybody wasn't learning together, everybody was discovering new things. I guess it took three or four weeks to actually get a fair idea, an overview of how Director works. but the more and more we used it, the easier and easier it became, you know, get into things fairly easy.	SuperCard dismissed because the team thought it was a HyperCard variation and not capable of sound/video requirements. Authorware Professional was considered but dismissed because of the bad experiences of their supervisor. Criteria were: Ability to support QuickTime. and allow use of digital video. The ability to include sample sound and speech Capacity for reference to code.	
What was the medium of distribution and why was it chosen	3	DR5	Why did you choose CD-ROM as your medium of distribution?	*Why did you choose CD-ROM as your medium of distribution? Simply because it was so big you wouldn't get it on anything else?* As opposed to disks? CD-ROMs. Everybody's going on about them at the moment - buzzword. Huge amount of storage which we needed for QuickTime movies. Something like the actual movie itself was only a couple;e of megabytes, and the QuickTime movies were about 450 megabytes worth of information. So if we wanted to do that we were going to need something huge. CD-ROMs are also very cheap, very cheap to produce in mass quantities as well. Guess they're more reliable than a floppy as well. *What about the availability of CD-ROM drives in schools - was that a worry?* Yeah. That's when we wrote all those letters to the school, that's what we were asking them. It was surprising how many schools actually do have a CD-ROM. OK, maybe only have one, but they have a CD-ROM player. Which was quite surprising for us as well - I didn't expect, so just kind of guessed that we were going to go for the CD-ROM.	Storage for digital video. Costs of production. Reliability.	
What authoring and programming tools were used?	3	DR6	Please give details of the authoring or programming software used in the project	–	MacroMind Director version 3 (prototype), version 4 (second phase). (GH01, G029, DR4) Only one copy available, which proved constraining at times (DR7)	

What ancillary tools (editors etc.) were used? [If prototyping was used] what prototyping software was used?	3	DR7	Please give details of all ancillary tools used in the project [name should do in most cases]: *Prompt if necessary from list below* video editors: sound editors: text editors: chunkers/file splitters graphics software: prototyping software any others? [If prototyping was used] what prototyping software was used?	*I've got a little question about ancillary tools. What sort of ancillary bits and pieces did you . .* - extras and optional tools. *Editors and what have you* Adobe PhotoShop, for, in Director there's a basic paint package, I don't know if you've used it, it's a bit like MacPaint or something: you can do bits and pieces. PhotoShop - the first time we used PhotoShop was trying to make fancy backgrounds and backdrops and sort of coloured effects and things, but then when we went on to doing the photos and stuff, we took photos, we originally started scanning photos in [?] using PhotoShop, they had to be re-sized cropped to fit in the allocated space, you know, so we used PhotoShop a lot for resizing, and changing the amount of colour information in pictures, because basically there's 256 colours on an LC2. Later on we used Adobe Premiere for putting together our own QuickTime movies, cutting and pasting. MovieShop - don't know if you've heard of that. *I've not come across that one.* Who is it that does MovieShop? *It's a bit like Premiere is it?* Yeah. That had all sorts of different compressions for when we were actually recording the videos we'd shot off the video recorders onto the machines, we used either MovieShop or Fusion Recorder. This lets you actually capture it, and compression algorithms, you use different ones which have got advantages for animation and short footage. *What about sound, what did you do for sound editing?* Sound editing was done with er SampleEdit, it was called, it was just a basic sample editor, have you heard of it, *I'm not sure* I'll check it later on, I'm not too sure. It was just a basic sample editor. Played it and recorded it. And then we had obviously word processors, Word or something, Microsoft Word, Claris Works. Then we had, we used Simple Player a lot, you know, that Mac Simple Player, just for playing QuickTime movies a lot, rather than loading them through Director and playing them, just when we were - the thing with Director was we only had one copy of it, so only one person could only work on it at a time, so if somebody was trying to do other things you know, that sort of, that's why we used Simple Player. *What about prototyping, did you use anything else for prototyping?* Mmm, no, it was done in Director 3. I don't think we used Premiere at that stage, because learning a big - Premiere was a big package, just like learning Director - I think we just did a simple copy and paste from to put together our movies; I don't think we edited them at all, it was just kind of 'yeah, we can get them to play, and they look - we can have them this size, anything bigger's not going to work, you know, [inaudible]	A wide range used, including: PhotoShop, (G028) Simple Player, MovieShop, Fusion Recorder (G030), Sample Edit, Microsoft Word, Claris Works, Premiere (G027). PhotoCD (G026). No use of prototyping software (DR7b. G032 - feasibility report).	

Design resources?	3/5	DR8	What design resources were available to you?	_	No specialist design resources. Video equipment and some stock footage available from the Media Department. Guidance from Phil Gray on design issues (GC4).	
Design decisions						
Where was the subject matter/information content derived from?	3	DD1	How did you obtain the information content?	*More incredibly boring questions. Design decisions. Before - when you actually got your information content, you used the original HyperCard thing, where else did you get information from?* In the prototyping stage it didn't really matter what information we put in, we could churn out the Bible all over the pages or whatever, you know what I mean, at that stage it didn't really matter. It was just organising text, what fonts would be, how it was going to look, would we have windows popping up and pooping away, more sort of the presentation in the prototyping stage than the actual content. When we actually went on to design the system, Catherine Lyons and Phil Gray as I said came into it a whole lot more; she's information officer, so she had books and books of information, and as the sections evolved, if the information wasn't there, then she would write the information, or there was like, say the section on the pubs and clubs, we wrote that sort of stuff. She did more of the formal writing of the details of courses and stuff like that, but to give it a more 'studenty' feel, I suppose we sort of wrote the unions section, and the pubs and clubs, and the 'what there is to do in the West End'.	Stakeholders, especially Catherine Lyon, who actually wrote a lot of the information.	Felt the interviewee was looking a little bored. . . Interesting that they felt the information content didn't matter.
How were design ideas generated?	3	DD2	How did you obtain your design ideas - [eg. brainstorming, experience of similar software]		Experience of similar software, for instance the original stack (which seems to have suggested what not to do), the ITN CD-ROM, from Richard Cooper, and Phil Gray. (G042,	

Was prototyping used?	1/3	DD3	Did you use prototyping	*What kind of prototyping did you use; what was your approach?* What do you mean, like rapid prototyping, heh, very rapid. *Yeah, horizontal vertical whatever, I mean, never mind the jargon!* I don't know, we didn't know anything about how you should go about prototyping a multimedia tool or. . . it was something new, it was something that I mean everybody else who was doing their own projects and that were doing something quite special that wasn't taught at that stage so. . Method we had for prototyping. Basically it was just trying to - basically the first thing was trying to decide on the structure, how it was going to be organised. That's when we were playing about with positioning and things, trying to find out what was looking good, how we could fit in navigation and control to actually get round the system, and a lot of it was that, actual content didn't play much of a part. We knew we would have to fit in a few QuickTime movies here and there, a few pictures, a bit of text. It was basically top-down. *So you do structure, put down a skeleton, if you like, without a lot of detail.* That was one of the first things we did when we started using Director, laid out all the different pages and all the sections, and filled them in top-down. We didn't in the prototyping stages, we only did a few sections. [end of side 1] Basically how we could move about from sub-menus up to menus, and once we were in sub-menus how we would organise going between pages if there was information to fit on the screen. It was more organisation and control, rather than content. I guess our prototyping was top down. *Right. You'd done your storyboards by this stage had you?* Yeah. *Did you show your storyboards to anybody?* To our project supervisor, just to make sure we were doing the right sort of thing, you know. *You didn't show them to any prospective users, any other people you found lying around?* No, no we never. It took us, we didn't actually get hands-on experience with Director for a long time, so all these documents were done in advance. We knew we were going to use Director, but before we actually got, or we figured out what Director was all about and actually got it working. So it's kind of difficult, because you've got this sort of like an information gap between knowing what you want and not being sure what Director will actually do. There was a lot of ideas thrown out and new things: 'Oh, maybe we could do this, this is quite good'. *I guess once you get hold of the package then certain things suggest themselves.* Yeah it's expectations of 'Can we do this?' 'Oh this would be a good idea', 'Why don't we make it do this as well'. And someone's been reading the Director book 'Oh we can do this as well?' Oh no! I want to [inaudible]. Just never-ended	Constructed a first prototype as part of a feasibility study. This was replaced with a later version. This does not fit easily into the common prototyping distinctions of rapid, evolutionary or incremental prototyping. combines elements of all three. A partial prototype was built, without extensive evaluation taking place. This was largely rebuilt using the same basic concepts and many of the interface features (G086a,). Much work done on paper due to lack of availability of Director at first (this section).	No storyboards seen. Pity.
What kind?	1/3	DD4	What form did this take? How many iterations?		As above.	

How was the information structured?	1/3	DD5	How was the information structured? [See product also]	–	Largely a shallow hierarchical structure, but with a rather complex cross-linked section dealing with course options (GH01, G043, G092). Shown, and quite possibly this reflects the way the were designed, as modularised units and sub-units in G092.	
Why was the information structured in this way?	1/3	DD6	Why did you choose the structure that you used?	–	Rationale to simplify navigation (G092).	
What design techniques and tools were used? (storyboards etc.)?	1/3	DD7	What techniques and tools did you use to work out your design?	–	Modularised structure diagrams, eg G092.	
What user interface design principles and guidelines were used?	1/3	DD8	Did you follow any user interface design principles or guidelines? Which ones?	Did you follow any user interface design principles anything like that? Just common sense. Well, I don't know how much of it is common sense, we just seemed to have a good idea of what we wanted to do, separation and things, we wanted to have control, we wanted to make things look as if they could be pressed, want the static things to look static, a lot of ideas.	'Common-sense' - but this isn't strictly true, as document G012 shows - a set of user interface and general software design principles.	
How useful were these principles etc.?	2	DD9	How useful were these ?	–	References to principles appear in such documents as the requirements analysis (D005). The principles concerned are somewhat vague, for instance 'design for a non-modal system', but appear to have been followed to some extent. Note that they are also aware of the problem of 'cognitive overload' despite its non-appearance in their list of design principles (G012).	

Navigation	3/4	DD10	How much attention did you pay to navigational problems? What efforts did you make to overcome these problems? [evidence of system]	*How much attention did you pay to navigational problems? People talk about hypertext and hypermedia, always going on about navigational problems. . .* Wrestled with them when we started doing the prototyping. Control was basically, it was hierarchical. You could move down menus and down menus, down layers, and once you got to the bottom it was pages, back and forwards, and we had a few ideas and they got thrown out, but as I say when you're in a sub-menu, or some sub-menu, you don't really know where you are relative to the rest of the system, so, took a sort of path-name approach, which we used along the top, top of every screen, which showed you where you were. I think in the HyperCard stack Jill Russell had a 'where am I' button, which came up with a data map, a network of where you were, but we went from a sort of had the highest menu, then the next menu you went to, it was like a path-name, right down to where you were. Didn't really, didn't really show you where you are in relation to other sections but you could see where you were in relation to the highest point. Other control things. For going back and forth between pages we had like left and right buttons. Goes to next page, or you can go back a page. Once we got to the end of the page we had kind of an up arrow which would sort of suggest after you finished this page then you had were going to go back up, but I think we took that out because it was a bit inconsistent with the controls being just left and right - people aren't really very sure. *Yes, a bit confusing I suppose.* And then we had like greyed-out arrows and things to try and suggest that you couldn't - *- that these features weren't apparently available*	Some awareness of the issue. Response to follow a hierarchical approach, and to take a path-name approach to identify each location. Rather computer-buff oriented.	
Designing nodes & links?	3	DD11	How did you decide how much information to put on each screen? What influenced the choice of button types?	*How did you decide how much information to put on each screen?* Judgement ourselves of what we thought wouldn't overload the user, what not - *- not too much , not too little* We only had a certain amount of space to work in obviously, but text had to be a certain size, you can't make it too small, so that sort of limited how much text we were going to put on a page. And the thing we wanted to keep the user interface consistent over the whole system so we'd sort of defined like different zones of the screen, different windows that would - OK, we're going to have a text window, we'll have - this window can be a picture, this is going to have e a sort of section at the bottom which depending where you are it could be a course map, or just another picture and stuff, so we're kind of limited in how much we could put on the screen. -	Assessment according to likelihood of cognitive overload, space available, text size, principle of consistency.	

How were access methods decided (apart from hypertext links)?	3		DD12	What influenced your choice of information access methods?	*What about your choice of information access methods. I seem to recall you had some -* Short cuts and things? *Yeah, there was, well there were little hypertext shortcuts, I'm trying to remember what -searching, you had a -* - We had a question mark on the main menu, which was the A-Z. That was one of the very late things we did. *That's right* When we were doing the project, it was all talk about, 'well we'll have a search engine running in the background where we'll have a form and you'll be able to type in what you're looking for and things. But it proved a bit - by the time it took us to actually develop the CD-ROM, it was, there was no time to actually go about doing that, so we had to try and conjure something up with Director. *Yeah, I guess in Director it would have been quite hard to do that, whereas in HyperCard you just appropriate the 'Find' command to do something like. . .* Mmm yeah absolutely, but as I say we didn't know anything about that at that the time, so we had to come up with an idea. I th nk that's the only way you can really search for things, but the way we structure it and then use words like they should have been, should have been a more obvious way of find things, by separating it into distinct sections, the University. Yeah so it became it might be a few things you'd say 'oh is that [inaudible] the University, the city, or the Department. *Right.* The only problem with that which was then we were restricted to the user for what they were going to look for, so they had to know what they were going to look for, and er that sort of thing, but enough [?] pages first, and that well, then again that was our judgement of what the key word was going to be, what they're going to look for, so maybe they'd wanted to look for, I don't know, drinking or something, mm, obviously that they wouldn't find that, they would have to look for pubs and clubs. *Yeah, yeah.*	Key factor here was technical and time limitations, which ruled out a dedicated search engine in favour of the more limited facilities possible within the Director authoring environment, where it was necessary to anticipate in advance the kinds of things users might wish to look for.	Interviewee missing the point a little. . . Hmm. What *is* he talking about?
Design of non-textual components - was this integral, or added on?	3		DD13	Was the design of non-text elements includec in the original design or was it added on?	–	This appears to have been integral - the early documents refer to 'multimedia' (eg. G005), and the consideration of software appears to have focused on the multimedia handling capacities of the various packages(.	

What influenced choice between authoring and programming?	3	DD14	What considerations made you choose authoring/programming [delete according to subjects]	*I think I can probably guess what the answer to this question, but what made you choose using a sort of broadly speaking authoring package like Director, as opposed to programming something?* It's more interesting, I think. *Yeah.* There was obviously the personal aspects of the project as well, when we were doing the prototyping and we wanted to do something interesting, we wanted to do something different. Sick of hacking out code, we'd been doing it for since -. This was an opportunity to design something which was interesting and it was more closer to real life systems than, and myself and the other people were more interesting in developing information systems than being software engineers and hacking out code, so we were wanting to actually get hands-on experience in designing this sort of thing. *Yeah, makes sense.* I don't really think a programming language would have been appropriate for this type of thing. It would been - I mean the framework that Director provides you is, you know, is fairly substantial, there's enough there to - it's not great but - *It makes the video and sound handling a lot easier* As opposed to us writing actually trying to code operating our own framework, including videos and such *That could have been quite a nightmare really.* Yeah, wouldn't have wanted to have done that. *Yeah, sounds pretty hard.*	Personal interest - 'sick of hacking out code', interested in information systems as opposed to software engineering. Also simpler to use the ready-made features of Director than trying to write routines for everything from scratch.	Paul is speaking for the others as well, so this may be questionable.
Evaluation with users - what methods were used and when?	1/3	DD15	What evaluation did you do with users? What method(s) did you use? When did you evaluate?	*Who did you test the system on? You've already told me about . . .* We had an open day. *Oh right.* We had an open day, I think it's September, September or October, the usual open day, we have people coming up to the Department. At this stage, this was after the prototyping and after the third year, we were actually designing it, and it was close to being finished, so we had a couple of CD-ROMs burnt in, which was sort of like, I suppose it was sort of a partial system then, it wasn't totally finished. Students would come into the Department, we'd have machines in the Department with the CD-ROM running, and we collected, I think we collected about a hundred questionnaires, which we'd got them to fill out, which was 'how did you find the system', 'what would you like to be in the system', 'what annoyed you about the system', they were a lot of help in that as well, 'cos there were things that we were taking for granted. We thought you know I mean this should be straightforward, this is, this is what people want to know, and er quite a lot of interesting things came out of that actually.	Main evaluation via a questionnaire and protocols conducted using prospective students at the Departmental Open Day, with the aid of psychology students (DD15a, G013-015 - results, questionnaire and protocol tasks).	

Cont.				*Did you try doing verbal protocols, and sitting people down?* And get them to carry out tasks? *Get them to tell you what they were doing?* Mmm. . . . . that, that was done on open day, on an open day we didn't just sort of open the door and let people run in and play with it, we were wanting, we were, there was people there while they were using them, and there was a couple of psychology students in the department, who were doing sort of HCI type things, and they were drafted in to do a sort of behavioural watch I suppose, and watch what people were doing, and er yeah there was, and we had a task sheet for them to carry out, as well as just playing with it. They had to run through the tasks and find this thing, find that thing, and yeah there was actually. We had questionnaire sheets and we also had these written sheets that these people had done, saying 'oh they had troubles with this, oh they clicked on the wrong button, they went back, they were confused', and sort of asking them while they were working with them. *Sounds quite good actually.* I'd forgotten about that. I wasn't actually there for the open day.	Also some informal evaluation via schoolchildren brought in by Catherine Lyons (GC9).	
What evaluation criteria were used? [Were they based on the specification of requirements]	3/4	DD16	What evaluation criteria did you use? Did you base them on the specification of requirements?	*What about the criteria for evaluation? Was that sort of based on your specification of requirements?* The requirements are sort of fairly general in the sense that it has to be easy to use. How can we judge how easy it is to use and such things. The criteria for assessment. Basically sitting people down with it. Making sure it wasn't major problems. People are happy to use it. Obviously we couldn't please everybody, but just trying to get a an overall feel for it; it was useful and purposeful. Think that was the only way we really went about it. *Right, yeah. After you'd done - well, you say you'd evaluated fairly late on.*	The requirements document (005) actually has some quite specific points, but the evaluation tasks and questionnaire do not directly correlate with these. Tasks and questions focused on navigation issues, picture quality, movie quality , effectiveness and value as conveyors of information, use of sound in gaining users' attention, general graphical appearance of the system, general effectiveness of the system, and suggestions for improvements and general comments on the system.	Paul's comments don't really do them justice - he obviously only has a very general memory of this part.

What authoring strategy was chosen - start from scratch or new structure?	3	DD17	Did you decide to design the system from scratch, or did you adopt structures found in existing documents or systems?		Started with the old HyperCard stack - they didn't like the structure of this very much, and changed the emphasis and dropped some elements altogether(GC10a).	
At what point was a detailed system design produced? (if at all)	1/3/4	DD18	Did you produce a detailed system design? When?	*At some point I presume you'd produced a sort of detailed system design. When did that actually happen?* What do you mean by a detailed. . . *Well, sort of, I suppose a blueprint based on the prototype and what have you.* I don't think we actually - after we started, after prototyping, started designing, we didn't do any documentation at all really. *Right* We were working on it, we were deciding on a framework, it wasn't like put it down, said this is it, this is the standard. I don't think we actually wrote it down, or kept it, or documented it, it was kind of 'this is the way it's going to be', type thing, so I don't think there was a sort of, we didn't actually lay down a document saying 'a formal - this is what it's going to be, 'cos like, suppose we didn't really have a customer, we didn't like, it was just a general, we were just throwing it out to people, we couldn't say like 'we promise you it'll be this, that and the next thing, because - *Yeah 'cos it's not like you made a contract or anything.* Yeah, so we, I don't know, maybe we should have. *Do you think it would have made life easier if you had?* Don't know. I don't think so. I mean we could have - *Sometimes, I've found myself, you sort of you get going off the rails, you know, kind of thinking, 'well yeah let's put one of them in, let's do some of that. Let's have some of this', then you know it's kind of handy to have a design which says 'hey, this is what we're supposed to be doing, along with your requirements specification.* Yeah, I think our design was very open in that way, it wasn't restricted. We did, well, had the basic framework, and what was going to, how it was going to be organised, but we did have ideas: 'oh we could have this, we could have, we'll have a special button that does this, and this pops up, and these sort of things were suggested and implemented; I guess we decided ourselves whether, whether it was useful or whether it spoilt the consistency or. . . . basically just trying to think up things ourself, and throw them in, so it was kind of flexible, it was, most things we didn't change, I think we kept it consistent, overall. I don't think it was a problem , like we didn't it - restrict ourselves.	The documents suggest that a reasonably detailed design was produced for the first phase. A user interface design document was produced (G007). Layouts of the hierarchical structure were drawn up (G092). No detailed design for the second phase. Used the earlier work as a basis and extended it. A generally loose and flexible approach (DD18).	

What form did this take? [eg scripts, storyboards, flowcharts or other structure diagrams]	1/3/4	DD19	What form did the system design take?	*What form did this system design take?*		
Construction						
What was the sequence used in building the final version? [this should ideally be a detailed account at level of preparation of text, graphics, video etc., construction of screens, backgrounds etc.]	3	C1	What was the order of construction in building the final version?	*That's good. When you came to actually doing the final construction, what was the sort of sequence of building things.* Concentrating on the Department section first of all, because we wanted to get it out the way, 'cos we knew it was going to be the biggest, it was a number of people working on it, so each person concentrated on a different section, 'cos it's basically up to them to do as much as they could on each section. If the Computing Department section there, that changed so much over the time that things were done and they had to be changed and redone and the actual guided tour section there was a lot of scripting in that section, which was causing a lot of problems, so that took a lot of work, a lot of time to do, and there wasn't just one person working on that, it would have been a bit unfair getting them to do that, so er mmm. Basically I think everybody else who was on the team concentrated on the Department section, and I sort of polished off the city section, the University section, because they were like getting involved with the heavy scripting and the coding stuff, and I guess - and I'd mailed Phil Gray and said 'well if you need help gathering information, then, then I can do that', so I was taking photographs, I was putting in text, I was doing the sort of information capture thing rather than the programming, so I was sort, I could get those things out of the way fairly quickly. They were just you know again [?] a bit tedious but, if you get them done, no problem at all. *Did you, sort of, when it came to the actual putting together things, did you work from sort of backgrounds upwards?* When I was putting together? *The actual, the Mac - using Director, did you work, did you sort of build your backgrounds, get your backgrounds sorted, or er. . .* How did we compose the different screens and such?	Sequenced by section. Department section first, then City and University sections worked on concurrently with the scripting for the Department section. Began with backgrounds and then added in other elements. Found getting the layers difficult with Director - much shuffling about. Difficult to design these elements in advance, and an ad hoc approach is required. (C1). Structured the document into several sections made from separate Director movies' linked together. Made skeleton structures for each module, then filled them in (C2).	

				Yeah er, how did you sort your - Mmm, at first we didn't really, didn't really think it mattered, but once you know, get to know Director, you realise that you've only got a certain number of channels to work with, and things have to be behind things and that, so. Back-backdrop was there from the start, because when we prototyped it looked very empty, there wasn't a lot there. So that was always there. It was so hard using Director because we had to put up a framework, so say we had windows which were going to hold information and things, and then later on we'd realise 'oh well we'll have to put text in the window, with the picture, but the picture's just a bit too big, and it's not rounded at the corners, so we actually have to put it behind the frame, and make the frame transparent. things like that; we had to shuffle about so much things in the actual, er the Director, organisation. *Do you think that there's a sort of logical sequence that you ought to use? Or do you think that it's always a sort of ad hoc process?* I I think that it's ad hoc and you have, you know if you put up your framework that it going to look like windows, it's going to be stuff that going to go behind, but you don't know for sure how much, how far down, how many things are going to go behind [macs[or in front of [macs], because we had buttons on top, the top of the interface in some sections, and if you actually got a look at the Director score, the Director score, with how it's all organised, you'll see that certain windows may be on the same channel for a while, but when you get into a different section you'll be down here because there's more to get in behind it. It was actually quite difficult in that sense.		
How efficient was the construction sequence?	4/5	C2	With hindsight, do you think this was the best way?	*Yes, I suppose it was quite awkward. That was. . . . What would you do next time?* What would we do next time? Ha. Would we do it again next time? I don't know. I mean, there's the opportunity there for the Department to do one every year and keep it up to date. Be a lot easier this time round not starting from scratch, but there was a few things that we did this time round that are a bit. . . could have been done a bit better. For one, do you, do you know much about using Director, just bits and pieces?. *I've not used Director for any sort of major thing, I've used it -* Yeah well you've got the idea of movies, and you can have separate movies linking to other movies, so when you click from movie to movie, it has to reload the other movie - it has to load movies to move in between. When we were developing the skeleton for the menu structure, we started filling it in and - I don' t know how this happened but somewhere in the design decided that sub-sections of the main menu were going to be contained in their own movie, as opposed to having just one great big movie with all the [?], we were going to have like a skeleton which was a [?] kind of thing which would load the relevant pieces like that. But once we'd introduced backing music and things, it became more obvious that switching between sections was causing a problem, because the music would stop. It does that in some places on the actual CD-ROM, so I guess - I mean we had sort of an organised approach when we did it, but I think knowing what we do now, we could have organised it a lot better.	Would reorganise the 'movies' (C2).	

How difficult was the timing of time dependent elements?	5	C3	How hard did you find the timing of time dependent elements?	*That's one of the questions I've got here is, how hard did you find timing the time-dependent elements? It sounds like a nightmare!* Do you want me answer that one? *You can do!* The time-dependent things. Well, basically there was just QuickTime movies and sounds. Nothing else was time-dependent. I mean Director does let you have - we went for a HyperCard frame-by-frame approach, with objects moving on each frame, but that was about it. QuickTime movies. We had a lot of arguments about how we were going to control QuickTime movies, whether we were going to provide the standard Apple interface, that's you know bit of a, where you rewind this slider bar and such, but we decided that we really didn't need that and offering that sort of control was more confusing to the user: they could find themselves in the middle of a movie when they came back from a page and that, and we wanted to keep it sort of standard, so we adopted this approach where if you left the page and you came , it would automatically go back to the start, so you weren't confused if it came to the end of the, end of the film. I think it looped - not so sure - think it looped round. So basically we just offered them a 'play', a 'stop', and did we have a? Aye, we had a rewind as well. It was a 'play', a 'pause' and a 'rewind'. We, we were just going to have a 'play' and a stop' but we decided against that because we didn't want to the user to come into a movie thinking they were at the start and pressing play when they were actually in the middle. *Yeah.* So our default on the times was when we came back to movie it would automatically rewind to the start, start playing. Music timing: well basically we just patched like three or four different tunes - great tunes they were! - there's actually one on the TV just now, on an advert! *Yeah?* It's awful. Concatenated them all together. Put them on loop and let them play over and over. Just at their own pace, what, what the system could handle, so we had to, we offered a 'turn the music off' option, which is something that's maybe neglected in many annoying systems, but we thought this would be a good idea. The fact that we had to use it so much, we were wanting an on-off button when we were prototyping! We wanted to be able to switch it off; that was part of it.	Caused a lot of arguments with regard to how the user should be able to control the video sequences, in particular making sure that videos reset to the beginning. Music relatively limited, and fairly straightforward. (C3)	

What system documentation, apart from designs and specifications, was made?	3	C4	How did you document the system?	*The documentation. What sort of documentation apart from the designs and the specifications did you have in the end?* Documents? *Like a final description of the product, anything like that?* Like a user manual type thing? *Mmm. Apart from the thing that was in the -* - in the CD-ROM? No we didn't. We only had all the design documents - I can show you them later on. It's a whole host of them, right through, feasibility study, report, project report, but none of these were actually going to be used with the CD-ROM, that was more for our guidance. We felt that, I mean, one of the things we were going to do, we were going to have an intro. at the start when you first loaded it up, which would show you how to use the system. But we felt that it wasn't necessary, because we designed it to be as simple as possible. There was a few things there, and there was a few things inside the CD-ROM cover, which I don't think's really explanatory itself, it's really quite messy - 'Click here for . . ' it's not really too obvious its it? *It's quite complicated isn't it?* Sort of basically the idea was if it was green you could press it. I don't know. It was good to some feedback on it and see how people actually found using it. I know Phil Gray's getting a lot of mail and phone calls of people having problems. I suppose he'll be considering all these different aspects and considering whether he wants to do a new one or improve any such. We were just left to it.	Extensive but unstructured approach to documentation (G001-153). Some elements well-documented, eg. fairly detailed log for first phase of project (042). No master documents for files, links etc..	
When was documentation for users prepared?	3	C5	When did you prepare the documentation to accompany the product?	–	A first user manual to accompany the prototype was prepared in June 1994 (G040).	

How was the system debugged [incl. procedures for link checking]	1/3	C6	How did you debug the system? How did you check the links all worked?	*How did you debug the system?* Open day helped a lot with that. People doing things unexpectedly we hadn't done. It was crashing left, right and centre! There was things there that shouldn't have been there! Open day spotted a lot of bugs. As I say, letting other people play with it, like Phil Gray and Catherine Lyons, different people in the Department who were wanting to come and play with it. They were obviously looking for bugs. They find - found a few as well. I suppose different- four of us in a team each doing a different thing. Occasionally we tended to wander about each other's sections and see what was happening, see if we could find that didn't work. So that was quite good there because we were looking more at the sort of scripting and things. *That's quite a good argument for having a division of labour then, that you do get -* - Competition type thing. So we were looking at more technical things when we were doing that, and there was, a group of, group of staff as well down the end of the Department who did a lot of the last checking, just trying to rip in everything they could think of. People with too much time on their hands, I guess, sitting playing with it. That's really how we went about debugging it. *Just general questions. What do you think were the main problem areas? You've mentioned several of them already - just to summarise.* Problem areas. I don't think - gathering the information was OK. QuickTime movies caused a lot of problems. Actually getting the compressions right. What we - I mean we were prototyping on a, we were designing on a Quadra, a really fast Mac, and some of the things maybe you'll never notice this would cause a problem, so we actually started using it on the LCII, so that we wouldn't 'oh no, this isn't going to work after all this!' , so QuickTime movies we had, had to compress them properly, make sure they could be pulled off the CD-ROM fast enough, you know, there was a lot, a lot of hassle with that. I mean, when we did a, burnt a few CDs for the open day, we had a few QuickTime movies on them, and they were playing fine and no problems at all, and then later on, near the end of our actual going to do it, we did, we burnt the whole thing again onto a CD-ROM and we were playing QuickTime movies off the CDs, and they were just jumping all over the place, and, and not too sure what was wrong. That caused a bit of a problem for a while and I think it ended up with being the Director movie getting so large it was causing problems. So we had to compress them properly, make sure they weren't long enough. Other problems again were just I suppose allocating work, getting access to it and getting it done in time was a problem, actual team-work. The organising of the score [inaudible]. Version control as well, come to think of it. People coming in and out, saving it, you're coming in, not knowing which one, which one's the most recent one. People not - you wouldn't think that would be a problem but it was a problem at some stages - you never know which one you're working with, and . . . *There's a case there for kind of compartmentalising things and putting them together. . .* Yeah, there was definitely problems with that [inaudible].	Testing at Open Day, allowing stakeholders and other members of staff etc. to play with the system. Also the separation of the team into different sections enabled a fresh view of each other's sections that showed up faults (C6a). Made lists of some of the problems and un-implemented items, eg. G069, G104 and G113. Some evidence of link checking - G104.	
What arrangements were made for future editions?	3	C7	What arrangements have there been for future editions?		Nothing known (C4).	

Success of the project						
How successful or otherwise were the methods employed?	2	SP1	Would you do things the same way again?	*I've nearly come to the end really. I already asked you, 'would you do things the same way again?'*		Not true - must have been getting confused with C2.
What were the problem areas?	4/5	SP2	What do you think were the main problem areas?	–	QuickTime digital video, especially getting compressions right, matching things to the lowest standard platform, and persuading video in a large document to work. Allocating work. Version control (C6a). Also problems with Director - dealing with objects and layers in the score (C2).	
Were the development team satisfied with the development process?	5	SP3	Were you happy with the development process?	*Were you in general happy with the development process that you chose?* Yeah, it's fairly successful. There wasn't any major problems with it. *That's good.*	Yes	
System acceptability						
Was the final system acceptable - to clients/commissioning authorities	5	SA1	Did the clients like the system? [ask team, client]	*Was the final system liked by the clients as it were?* I don't know, you'd have to ask Phil Gray. *It's hard to tell, isn't it.* I don't know, er, initially we were impressed with it, we thought 'this is quite good', but more, it's hard sort of to judge how people are going to react to it, and what we think people know, and what's actually out there, you know what I mean. Schools er, I don't know, schools are on BBCs or LCIIs, something like this, everybody's going to go 'Wow!' [?] we get pissed off if we have to use an LCII, you know what I mean? It's so slow and we're just so demanding now that we're at university, that we just wouldn't - seeing a HyperCard stack maybe schools who have just got a computer would think a HyperCard stack's great, you know, I mean it's hard to judge. So I guess you'd have to ask Phil Gray about the responses. But I think that overall it's quite impressive. It's a step.	Generaly quite pleased. Some problems with the interface design (informal discussion with Phil Gray.)	

Cont.				*I suppose that your summative evaluation's happening out there at the moment - Phil getting all this, these phone calls from people. Do you think they'll do another system?* There was talk of doing a University of Glasgow, a general one, yeah - what's the guy's name, I think his name's Mike Brown, he's head of advertising for Glasgow University. He's got millions of pounds to spend, I'm sure, so er so I think maybe he's wanting to do a general one. But again there's only so much you can put on. It would be like a prospectus, you would have to just see a little bit, it wouldn't be as in depth as this was, you couldn't like show shots of different departments, so I guess you're limited in that way. *How much space did that actually take up, as a percentage of your CD-ROM.* The QuickTime movies? *Well, yeah, the entire um . . .* Oh, how much of the 750 megabyte did we use. About 470. *To do the whole University would take quite a few CDs* Take quite a lot of time as well. Take over a year to do it, and by then you'd be doing next year's. You can imagine. *I know that thing that Jill Russell did -* [tape ends]		
Was the final system acceptable to users?	5	SA2	Did the users find the final system acceptable?			
Success of methodology						
Was the team satisfied with the methodology used (if applicable)	2	SM1	Were you as a team satisfied with the methodology used (if applicable)	n/a	–	
Was the development team satisfied with any principles used other than the methodology (if applicable)	2	SM2	Were you as a team satisfied with the principles used? (If applicable)	n/a	–	
Was the development team satisfied with the guidelines used (if applicable)	2	SM3	Were you as a team satisfied with the guidelines used? (if applicable)	n/a	–	

Was the development team satisfied with the model used (if applicable)	2	SM4	Were you as a team satisfied with the model used? (if applicable)	n/a	–	
Did the process (methodology) used produce the best possible system?	2	SM5	Did the process/methodology used (as applicable) you used produce the best possible system?	see above	–	
Did the methodology fit in with the team's way of working?(if applicable)	2	SM6	(If a methodology was used) Did the methodology fit in with the team's way of working?	n/a	–	
Did the methodology use the available resources efficiently (if applicable)	2	SM7	(If a methodology was used) Did the methodology use the available resources efficiently?	n/a	–	

Appendix 3c: Table GT03 - catalogue of documents, Glasgow case study

All files are Macintosh files. Size in k is approximate, based on 'on disk' figure in 'Get info' boxes. Dates also from 'Get info' boxes. Type is one of three: 'text content', 'working document', or 'documentation'. Titles are taken verbatim from the original file names. Descriptions of contents are this author's own.

No.	*Title*	*Size (k)*	*Type*	*Date created*	*Date last modified*	*Contents*	*Relationship to other documents*	*No.*
001	NewProjProbDef-long	5	working document	Wed June 8 94, 3.25 pm	Wed June 8 94, 7.29 pm	This is titled 'Third Year Group Project Problem Definition'. Outlines the problem, group goals, and then proposes a 4-stage development process.	Similar to 046, 047, 098	*001*
002	Existing Stack	50	working document	Sat Jan. 22 94, 6.03 pm	Thurs June 9 94, 12.44 pm	Comments on the existing stacks by Jill Russell		*002*
003	Contextual analysis	17	working document	Wed June 8 94, 10.54 am	Fri. June 10 94, 6.37 pm	Part of the requirements specification process: attempts to highlight the possible 'stakeholders' in the system and to discuss the scenarios within which the 'Glasgow University hypermedia document' will be used. And also suggest the tasks that each of the scenarios requires the system to perform.'		*003*
004	Finished User Analysis	9	working document	Thurs June 9 94, 4.10 pm	Mon. June 13 94	Identifies a number of user groups, apparently without any empirical research, including school leavers, overseas students, Open Day visitors, guidance teachers, departmental users, other universities, manufacturers, parents, and general visitors. An example of documentation prepared retrospectively!		*004*
005	Requirements	23	working document	Sun June 12 94, 9.27 pm	Mon. June 13, 94, 9.47 pm	General analysis on a 'who, what, where, when, how' basis. Includes general tasks the system is to perform, a system specification, a requirements specification, functional and non-functional requirements, etc.		*005*
006	QTM	7	working document	Wed June 8 94, 3.22 pm	Wed June 8 94,	Design rationale for QuickTime movies - reasons for inclusion, type of presentation and controls chosen, etc.		*006*
007	UserIFdesign	33	working document	Wed Feb. 23 94, 9.00 pm	Mon. Feb. 28, 94, 11.43	Examines sound, video, graphics, animation & screen layout issues. Gives some 'concrete representations' of layouts. No screen shots.		*007*
008	Design Decisions..Paul	10	working document	Wed June 8 94, 3.22 pm	Fri. June 10 94, 3.52 pm	Covers the QuickTime & some of the sound stuff from the UserIF design document (007).		*008*
009	Design Decisions FINAL	410	working document	Wed June 8 94, 3.22	Fri. June 10 94, 9.43 pm	Includes colour, screen layout & navigation, & QuickTime stuff. Some screen shots.		*009*
010	Design Decisions. Stuart. Paul	102	working document	Fri. June 10 94 , 4.20 pm	Fri. June 10 94 , 4.20 pm	Another version of the QuickTime & sound stuff.		*010*
011	Design Decisions.FINAL.FINAL	329	documentation	Fri. June 10 94, 4.55 pm	Mon. June 13, 94, 9.51 am	Looks like this may really be the 'final final' version of this. Smaller than 007 user IFdesign though. Includes screen shots.		*011*

No.	Title	Size (k)	Type	Date created	Date last modified	Contents	Relationship to other documents	No.
012	Design principles	9	working document	Fri. June 10 94, 10.40 pm	Mon. June 13, 9.53 am	Includes the following principles - where did they get these? - not referenced. Design for ease and pleasure of use Design for consistency Design for a non-modal system - Design for structural visibility - Design for flexibility Design for direct entry to information Design for extensibility Design for reversibility of actions		*012*
013	Evaluation	4	working document	Mon. June 13 94, 10.35	Mon June 13 94, 10.42 am	User evaluation tasks and follow-up questionnaire. Includes method of application. Date created makes one wonder if this was ever used for real - done at the same time as all the other report writing Guess the evaluation may have been a bit of an afterthought - or summative rather than formative! No conclusions - finishes with 'the following conclusions being reached'.		*013*
014	questionnaire	4	Final documentation	Mon. June 13 94, 10.38	Mon. June 13 94, 10.38	As above	As 13 but with spaces for users' replies	*014*
015	Evaluation Results	3	documentation	Tues. June 14 94, 3.55 pm	Tues. June 14 94, 4.30 pm	Very thin analysis of results of user tests.		*015*
016	StructuredAnalysisOf Prob	3	working document	Wed June 8 94, 6.28 pm	Wed June 8 94, 628 pm	A structured analysis of the problem that says there is no structured analysis, because it isn't appropriate to this kind of system - there's no data flows. Instead discuss the network diagrams that they've done. Unfinished.		*016*
017	CopmpSciStruc	7	working document -	Sat June 11 94, 6.31 pm	Tues. June 14, 5.16 pm	structure diagram		*017*
018	MainMenuDiagram	15	Working document - diagram?	Sat June 11, 94 6.00 pm	Mon. June 13, 94, 7.16 pm	structure diagram		*018*
019	GTheCityDiagr	7	Working document - diagram?	Sat June 11, 94 6.55 pm	Sat June 11, 94 6.55 pm	Structure diagram		*019*
020	GUniDiagram	7	Working document - diagram?	Sat June 11 94, 6.55 pm	Mon. June 13, 94, 7.20 pm	Structure diagram		*020*

No.	Title	Size (k)	Type	Date created	Date last modified	Contents	Relationship to other documents	No.
021	CousesAvailable	8	Working document - diagram?	Mon. June 13 94, 7.36 pm	Mon. June 13 94, 7.36 pm	Structure diagram		021
022	CS1A	6	Working document - diagram?	Mon. June 13 94, 8,15 pm	Mon. June 13 94, 8,15 pm	Structure diagram		022
023	glasgow today	6	Working document - diagram?	Mon. June 13 94, 8.40 pm	Mon. June 13 94, 8.40 pm	Structure diagram		023
024	GlasgowEnts	6	Working document - diagram?	Mon. June 13 94, 8.22 pm	Mon. June 13 94, 8.22 pm	Structure diagram		024
025	Untitled5	6	Working document - diagram?	Mon. June 13 94, 10.17 pm	Mon. June 13 94, 10,17 pm	Structure diagram		025
026	PhotoCD	3	working document	Mon. June 13 94, 9.22 pm	94, 9.22 pm	Short description of product & processing availability.		026
027	Adobe Premier	4	working document	Mon. June 13 94, 9.43 am	Mon. June 13 94, 9.45 am	Short description of product and, more importantly, how it was used in the project (not until late on in the prototype stage it seems).		027
028	PhotoShop	4	working document	Mon. June 13 94, 1.42 pm	Mon. June 13 94, 2.13 pm	Product description & details of use - including step-by-step procedure for incorporating PhotoCD images in Director		028
029	MacroMind Director	11	working document	Mon. June 13 94, 9.55 am	Mon. June 13 94, 10.02 am	Includes criteria for selection of a multimedia authoring package, the short-list of products - HyperCard, SuperCard, Director, and 'Authorware' - presume they mean Professional - and the pros & cons of each. Fairly detailed description of Director, the eventual choice. Finishes with a review of the then newest versions of Director and HyperCard, with regard to selection of software for the complete system, which was obviously considered possible at this stage.		029
030	Fusion Recorder	5	working document	Mon. June 13 94, 9.41 am	Mon. June 13 94, 9.45 am	Description of the product plus details of its use by the team with regard to image size and compression algorithm. Notes failure to get the sound to work.		030
031	SchoolAnalysis	6	working document	Thurs June 9, 94, 5.33 pm	Thurs June 9, 94, 6.08 pm	Letter sent to each of the nine Regional Council Education departments in Scotland to obtain an idea of the hardware platforms available in schools, and details of information received. Information partially complete - some regions had only sketchy knowledge, and recommended contacting individual schools. Strathclyde was contacted at area level - some areas had no info available. Don't know if they ever did any school level research.		031

No.	Title	Size (k)	Type	Date created	Date last modified	Contents	Relationship to other documents	No.
032	Feasibility Study	10	documentation	Sun June 12 94, 2.15 pm	Mon. June 13, 94, 3.24 pm	Places work in context of the INTERFACE group's needs, and explores the feasibility of taking the initial prototype and developing it into a full system which could be circulated on CD-ROM. Examines 'the status and achievements of the prototype'. Mentions limitations of the then current version of Director, and the possibility of going back to the new version of HyperCard (2.2). Notes that the video grabber was not successfully in use until the week before the project was due to finish. Details of difficulties in importing stills and sound. Possibilities of Mac to PC conversion. Assesses prospects for a full version - much of the investigative work still valid - could be reused giving time savings, but the prototype would not be suitable as a basis on which to build a full system, largely because of the way the 'score' in Director had grown incrementally and in an unstructured way. Cast members could be kept, but the score would have to be wiped and the structure better organised.		*032*
033	un ideas	6	working document	Thurs June 9 94, 10.40 am	Mon. 30 June 94, 4.17 pm	Looks like notes for 034. Some of the same material, but more odd bits and pieces of detail.	As 034 - shorter, more fragments	*033*
034	Unimplemented Ideas	7	working document	Mon. June 13 94, 9.51 am	Mon. June 13 94, 4.10 pm	Description of ideas and intentions not realised at the prototype stage, for a variety of reasons. Gives some indication of constraints.	Longer version of 033	*034*
035	Intro.Colin	6	working document	Mon. Feb. 21 94, 4.04 pm	Tues. Mar 8 94, 8.20 pm	Individual introduction/aims of project	One of three versions - 036 & 037 the others	*035*
036	Intro.Donald	7	working document	Sat June 11, 94, 3.40 pm	Sat June 11, 94, 3.40 pm	" "	See above	*036*
037	Intro.Derek	13	working document	Sun Feb. 20 94, 10.57 pm	Wed Mar 9 94, 9.31 am	" "	See above	*037*
038	Acknowledgements.CDS	4	documentation	Mon. June 13 94, 2.42 pm	Tues. June 14 94, 7.51 pm	Just what they say. Indicate contacts involved in project. Key figs - Richard Cooper Mark Sanderson and Iain Campbell INTERFACE, especially Catherine Lyons and Phil Gray, Mark Dunlop, John Patterson		*038*
039	Appendices[not]Full Document	120	documentation	Tues. June 14 94, 4.08 pm	Tues. June 14 94, 7.16 pm	Looks like appendices to the project report, for assessment - original aims + the INTERFACE report. Not complete.		*039*
040	FIN.USERMAN + notes	767	documentation	Sun June 12 94, 1.25 pm	Mon. June 13 94, 10.25 pm	What it says.		*040*
041	GroupLog1	30	working document	Sat June 11 94, 3.52 pm	Sat June 11 94, 3.52 pm	Seems to be another version of the 'don't use these logs' logs - presumably spruced up for assessment.		*041*
042	GroupLogPal	37	working document	Mon. June 13 94, 11.21 am	Check originals	Full log of the course part of the project, Nov. to June. Need to compare this to the short logs - see if they're consistent.		*042*
043	INTERFACE Report5	115	working document	Mon. Feb. 28 94, 12.02 pm	Check originals	This is on first examination v similar to the first report - includes a lot of material already presented there. Close comparison required.		*043*
044	-							*044*
045	ITN CD-ROM analysis	3	working document	Sun Jan. 30 94, 5.49 pm	Tues. June 14, 94. 7.38 pm	Short review of an ITN CD-ROM that the team looked at for initial ideas.		045

No.	Title	Size (k)	Type	Date created	Date last modified	Contents	Relationship to other documents	No.
046	NutherProbDefn	4	working document	Tues. June 14 94, 7.29 pm	Tues. June 14 94, 7.29 pm	Same kind of document as 001 - but doesn't mention INTERFACE, has slightly different goals.	Similar to 047, 098, 001	046
047	OldProjProbDef3	4	working document	Mon. Dec. 6 93, 3.06	Tues. June 14 94, 4.07 pm	As the title describes. Interesting that these problem definitions show an emphasis on machine capabilities. True computer scientists.	Similar to 046, 098, 001	047
048	choosing charlie	11	text content	Wed July 6 94, 12.29 pm	Thurs Sept. 8 94, 5.25 pm	A personalised description of some of the course options, featuring Greg, not Charlie		048
049	Transport - text 1	4	text content	Wed Sept. 14 94, 2.25 pm	Wed Nov. 9, 94, 2.02 pm	General description of the Glasgow transport network.		049
050	Entertainments - text 1	45	text content	Wed Sept. 14 94, 5.09 pm	Wed Sept. 14 94, 5.09 pm	Corrupt - can't open		050
051	CompScienceDump	227	Working document	Sat July 11 94, 6.19 pm	Sat July 11 94, 6.19 pm	Screen dump of prototype version of Dept section of the VOD.		051
052	Course pages	35	text content	Thurs July 7 94, 5.40 pm	Mon. July 11 94, 7.18 pm	Includes some early graphics.		052
053	-							
054	-							
055	sample projects	3	text content	Thurs Sept. 8 94, 7.14 pm	Thurs Sept. 8 94, 7.14 pm	Email message from Richard with details of project structure and previous titles.		055
056	single cs objectives	5	text content	Thurs Sept. 13, 94, 12.01 pm	Thurs Sept. 13, 94, 12.01 pm	Course objectives for Computer Science degree		056
069	Dear Donald...	4	working document	Sat July 23, 94, 6.08 pm	Thurs. 4.8.94, 11.41	A list of unimplemented stuff, difficulties and other ideas for Donald to consider and in some cases implement An illuminating document.	Same as 105, except that the dates are different - why?	069
070	Formatting	3	working document	14.9.94 - 3.41 pm	14.9.94 - 3.41 pm	Paul's note giving details of text formats to be adopted - don't know how this was agreed - plus request to Catherine re best format for raw text. Note limitations of the window format chosen - 13 lines, no more than 32 characters per line.		070
071	-							
072	Guide Colour	2	working document	Wed 13.7.94, 5.53 pm	Wed 13.7.94, 6.12 pm	Note on colour choices for members of the Director 'cast'. Includes note from Catherine.		072
082	Don't use these logs	2	blank	Tues. June 14 94, 9.08 am	Tues. June 14 94, 9.08 am	Empty file		082
083	Project log 11.11-24.1	15	working document	Mon. Dec. 6 93, 4.58 pm	Tues. Feb. 1 94, 4.35 pm	Fairly detailed log of first part of first phase of project - defining aims etc..		083
084	Log - Term2, 25.1 - 16.3	17	working document	Sun Jan. 30, 94, 4.50 pm	Sat April 16 94, 3.43	Middle of first phase of project. Good detail, incl. discussion of the INTERFACE report as well as the project itself.		084
085	Log -Term3, 19.4-3.5	5	working document	Wed, Mar 23 94, 3.40 pm	Wed May 4 94, 12.42 pm	Last stages of first phase section of project.		085

No.	Title	Size (k)	Type	Date created	Date last modified	Contents	Relationship to other documents	No.
086	Existing Stack	50	working document	Sat Jan. 22 94, 6.03 pm	Thurs June 9 94, 12.44	History & critique of earlier versions of guides by Gill Russell. This document appears elsewhere - modification probably tidying up for submission, not significant to the project		086
086 a	mainmenudump		working document	Sat June 94 3,28 pm	Sat June 11 94 3,28 pm	Screen dump of a main menu example from phase 1 - not sure which version yet.		086
087	Finished User Analysis	8	documentation	Thurs June 9 94 4.10 pm	Thurs June 9 94 4.10 pm	What it says - on the face of it. There are other versions of this in other documents, so these could be later versions. Done for assessment?		087
088	Structure discussion	50	documentation	Wed, 23.2.94, 8.45	Thurs., 24.2.94, 5.23 pm	Consideration of the various structural possibilities, main issue here being how to present the course contents information.		088
089	Front page	3	documentation	Sun Feb. 27, 4.51 pm	Sun Feb. 27, 4.51 pm	As it says, a cover page. Describes the INTERFACE report as 'Summary of structure, content, and user interfaces for Multimedia Prototype.'		089
090	Glas Proj Outline	6	documentation	Wed Feb. 23 94, 6.36	Sun Feb. 27, 94, 9.07	Proposed structural plan for the City info section, with outline of information content proposed.		090
091	INTERFACE Report	114	documentation	Mon. Feb. 28, 12.02 pm	Mon. Feb. 28, 12.02 pm	The full report. Needs checking for changes between the section documents and this, and any linking elements.		091
092	Layout	26	documentation	Mon. June 27, 94 5.04 pm	Mon. June 27, 94 5.04 pm	Discussion of structure and info content, including what is and isn't implemented at this stage.		092
093	MainMenu Info	6	documentation	Sun Feb. 27 94, 4,09 pm	Sun Feb. 27 94, 4,09 pm	Description of Main Menu page at this stage.		093
094	MainMenuCompDept -Csel copy	18	documentation	Sun 27 Feb. 94, 4.06 pm	Sun 27 Feb. 94, 4.06 pm	Info content and structure.		094
095	MainMenuCompDept -Csinfo	95	documentation	Sun Feb. 27 94 4.06 pm	Mon. Feb. 28 94, 11.57 pm	Looks like the contents of the INTERFACE report, maybe not complete, different version?	As 091	095
096	Uni-Proj-Outline Info	6	documentation	Thurs Feb. 24 94, 1.34 pm	Sun Feb. 27 94, 9.08 pm	More familiar stuff - this is the suggested campus information.		096
097	ITNanalysis	3	documentation	Tues. Feb. 1 5.14 pm	Tues. Feb. 1 5.14 pm	Empty document.		097
098	<ProjProbDef	5	working document	Mon. 6 Dec. 93, 2.06 pm	Mon. 6 Dec. 93, 4.08 pm	Aims and anticipated project stages	Similar to 046, 047, 098, 001	098
099	UserIFdesign Info	33	documentation	Wed Feb. 23 94 9.00 pm	Mon. Feb. 28 94, 11.43 pm	Includes sections on sound, video, animation and graphics, colour, screen layout, 'concrete representations', QuickTime screens, pictures/graphics. Nothing on wider issues in HCI. In main INTERFACE report?	In091	099
103	GuidedTour Structure	15	working documentation - diagram	Fri. July 8 94, 12.48 pm	Sun July 24, 94, 3.23 pm	Structure diagram		103
104	Query	5	working document	Thurs. July 7 9494 10.56 am	Tues. July 19 94 11.48 am	A 'thinking aloud type document - no details of writer or addressee. A discussion of the problems involved in linking and link representation, and some of the difficulties of scripting for Director. Illustrates the trade-off between structural and technical issues.		104
105	ThingsToDo	5	text content	Sat July 23 94, 5.59 pm	Sat July 23 94, 5.59 pm	This is a 'List of unimplemented stuff and other ideas.' No chronology. Good insight into 'work in progress'. Same as 'Dear Donald'.	Same as 069.	105

No.	*Title*	*Size (k)*	*Type*	*Date created*	*Date last modified*	*Contents*	*Relationship to other documents*	*No.*
106	Accommodation - text 1	8	text content	Tue Aug 23, 94, 4.24 pm	Thurs Sept 8 94, 5.25 pm	Text		
107	Accommodation - text 2 cl	5	text content	Tue Aug 23, 94, 5.03 pm	Wed Sept 14, 94, 3.40 pm	Text		*107*
108	EntertAmenities - text 1	8	*text content/*	Wed Aug 31 94, 5,28 pm	Wed Sept 14, 94, 1.15 pm	Text		*108*
109	History & Loc - text 1	5	*text content/*	Wed Sept 14, 94, 12.47	Wed Sept 14, 94, 3.35	Text		*109*
110	Libs and Museums - text 1	6	*text content/*	Thurs Aug 25 94, 4.05 pm	Thurs Sept 1 94, 3.19 pm	Text		*110*
111	Sports - text 1	5	*text content/*	Mon Aug 29, 94, 2.03 pm	Mon Aug 29, 94, 2.11 pm	Text		*111*
112	Sports - text 2 cl	5	*text content/*	Mon Aug 29, 94, 2.32 pm	Tues Aug 30, 94, 1.19 pm	Text		*112*
113	vodnotes 22 nov	3	working document	Tues. Nov. 22 94 10.44 am	Tues. Nov. 22 94 11.50 am	Notes of bugs and inconsistencies in the VOD, apparently from a meeting between the team and Catherine.		*113*
114	careyd	3	text content	Friday Sept 16, 94, 5.31 pm	Friday Sept 16, 94, 5.31 pm	Text		*114*
115	jstack	3	text content	Friday Sept 16, 94, 5.22 pm	Friday Sept 16, 94, 5.22 pm	Text		*115*
116	muffy	5	text content	Friday Sept 16, 94, 5.43 pm	Friday Sept 16, 94, 5.48 pm	Text		*116*
117	printezi	5	text content	Friday Sept 16, 94, 5.45 pm	Friday Sept 16, 94, 5.48 pm	Text		*117*
118	simonpj	5	text content	Friday Sept 16, 94, 5.29 pm	Mon Sept 19, 94, 5.29 pm	Text		*118*
119	wadler	5	text content	Friday Sept 16, 94, 5.17pm	Friday Sept 16, 94, 5.51 pm	Text		*119*
120	jej text	5	text content	Tues. Sept 14 94, 7.45 pm	Tues. Sept 14 94, 8.24 pm	Text		*120*
121	alison	5	text content	Friday Sept 16, 94, 11.14 am	Mon Sept 19, 94, 11.14 am	Text		*121*
122	guided tours signposted	26	text content	Thurs July 7, 94, 6.38 pm	Thurs July 7, 94, 6.46 pm	The beginnings of an interface for the guided tours section - text, plus some graphics to indicate the different options.		*122*
123	any guide	3	text content	Wed Sept 14, 94, 2.36 pm	Wed Sept 14, 94, 2.37 pm	Template for the following items.		*123*

No.	Title	Size (k)	Type	Date created	Date last modified	Contents	Relationship to other documents	No.
124	Fiona and Paul	11	text content	Sun Sept 11 94, 3.59 pm	Mon Sept 19 94, 11.03 am	Text		124
125	Greg	9	text content	Thurs Sept 8 94, 5.31 pm	Wed Sept 14, 94, 2.32 pm	Text		125
126	Norman	3	text content	Tues Sept 13 94, 11.38 am	Tues Sept 13 94, 11.38 am	Text		126
127	pauline	8	text content	Wed Sept 14, 94, 2.36 pm	Wed Sept 14 94, 5.30 pm	Text		127
128	sammy	8	text content	Wed Sept 14, 94, 5.12 pm	Wed Sept 14, 94, 5.28 pm	Text		128
129	simon	5	text content	Mon Sept 19 94, 11.05 am	Mon Sept 19 94, 11.24 am	Text		129
130	Stuart and Catherine	9	text content	Mon Sept 19 94, 10.02 am	Mon Sept 19 94, 111.20 am	Text		130
131	themes summarised	11	text content	Sun Sept 11 94, 6.08 pm	Sun Sept 11 94, 6.49 pm	Text		131
132	City of Glasgow	8	text content	Tues Nov 8 94, 9.39 pm	Tues Nov 8 94, 10.38 pm	Text		132
133	class exam	5	text content	Tues Nov 8 94, 9.15pm	Tues Nov 8 94, 9.17 pm	Text		133
134	cs2b.tmm	9	text content	Fri Nov 4 94, 5.18 pm	Fri Nov 4 94, 5.18 pm	Text		134
135	gu menu	5	text content	Tues Aug 16 94, 11.42 am	Thurs Aug 25 94, 3.13 pm	Text for menu items		135
136	-							136
137	3H projects	5	text content	Thurs Sept 8 94, 7.31	Thurs Sept 8 94, 7.31	Description of third year projects.		137
138	4H projects	5	text content	Thurs Sept 8 94, 7.38 pm	Thurs Sept 8 94, 7.38 pm	Description of fourth year projects.		138
139	sample projects	6	text content	Thurs Sept 8 94, 7.14 pm	Thurs Sept 8 94, 7.14 pm	A short description of four of last year's third/fourth year projects		139
140	single cs objectives	5	text content	Tue Sept 13 94, 12.01 pm	Tue Sept 13 94, 12.01 pm	Detailed course objectives for single Hons Computer Science.		140
141	cs3	9	text content	Mon Sept 12 94, 11.39 am	Wed Sept 14 94, 12.28 pm	Text		141
142	-							142
143	-							143
144	cs3h	8	text content	Mon Sept 12 94, 11.39 am	Tue Sept 13 94, 9.06 pm	Text		144

No.	Title	Size (k)	Type	Date created	Date last modified	Contents	Relationship to other documents	No.
145	cs3h+	6	text content	Tue Sept 13 94, 12.59 pm	Mon Sept 19 94, 11.38 am	Text		*145*
146	cs4h	8	text content	Tue Sept 13 94, 6.48 pm	Tue Sept 13 94, 9.06 pm	Text		*146*
147	cs4h+	9	text content	Mon Sept 19 94, 11.25 am	Mon Sept 19 94, 11.42 am	Text		*147*
148	csee 1&2	8	text content	Sun Sept 11 94, 4.48 pm	Mon Sept 12 94, 12.00 pm	Text		*148*
149	ese 3h	6	text content	Tue Sept 13 94, 6.28 pm	Wed Sept 14 94, 10.12 am	Text		*149*
150	ese 4h	8	text content	Wed Sept 14 94, 10.00 am	Wed Sept 14 94, 10.15 am	Text		*150*
151	more about SE (cs2a)	6	text content	Thurs Sept 8 94, 6.43 pm	Thurs Sept 8 94, 7.25 pm	Text		*151*
152	se 3h	8	text content	Tue Sept 13 94, 3.21 pm	Tue Sept 13 94, 7.26 pm	Text		*152*
153	se 4h	6	text content	Tue Sept 13 94, 4.15pm	Tue Sept 13 94, 7.27 pm	Text		*153*

Appendix 3d: GT04 - chronology of the Glasgow VOD project

This chronology is based on the evidence of the documents from the Glasgow case study combined with evidence from the interview transcript. Not all material is included here: material omitted includes copies of other files, and trivial material such as report covers. Some tasks that must have occurred are not mentioned here because no evidence exists for them.

Date	*Event*	*Evidence:*
1993		
November		
17	Project allocated.	G083
	First meeting with academic supervisor.	G083
December		
6	Problem definition - aims of project, anticipated project stages	G047
6	First project log commenced	G083
2	First meeting with client	G083
7	Meeting with John Patterson, discussion of possible elements for inclusion. Tentative listing of hardware and software requirements. Early speculation on merits of using HyperCard v. Director - problem with latter, no generally available license. Investigating feasibility of digital video.	G083
17	Discussion of possible scenarios for use. Examination of some existing CD-ROM packages.	G083
January 1994		
13	Temporary division of group into two teams, one looking at user profiles, scenarios, situation analysis, usability and human factors, and the other looking at the existing Departmental HyperCard stack.	G083

Date	*Event*	*Evidence:*
1994		
Jan. 14	Project aims clarified. Still clearly a feasibility study at this stage, even though this includes the building of a prototype. Prototype itself to be breadth rather than depth oriented, and to show the possibilities. Users, distribution, considered. Information source, the Dept handbook, and the existing Dept stack.	G083
18	Meeting with Phil Gray and Catherine Lyon. Aims fairly well defined after this - to "Investigate feasibility of an enhanced CD-ROM based departmental information source.". Considered information sources - the Dept handbook and existing HyperCard stacks - and the possibility of importing the texts from these and building an information retrieval search engine.	G083
19	Information gathering - hardware & software. Still considering HyperCard after seeing the new version, with colour, at an exhibition.	G083
22	Review of existing HyperCard stack. Found this severely dated in appearance.	G002, G086
23	Scenarios and user analysis completed. Structure of document under consideration, with main focus the course contents.	C083, C088
26	Preliminary decisions and suggestions. In particular, to build a prototype that included only a few examples for each facility but had a wide range of facilities implemented-"Breadth not depth". The 'actual nitty gritty of reading information in, etc. was to be done in the summer months.	G083, G041
28	Exploration of multimedia possibilities, looking at QuickTime. More idea generation eg. 'we could include doctored cut down versions of the demonstrations run on open day'	G084, G041
30	Initial log opened.	G084
30	Review of ITN CD-ROM.	G045

Date	*Event*	*Evidence:*
1994		
February 1	Aim at this point 'to have the whole structure and design finalised and implemented' by the end of the term' (mid-March). Looking at the feasibility of Director for the authoring tool.	G084
3	Liaison with Media Services re availability of equipment and stock video footage.	G084
9	Analysis of video obtained from Media Services noting useful elements.	G084
20	Introduction and statement of aims of project by Derek.	G037
21	Early structuring and user interface decisions.	G084
21	Introduction and statement of aims of project by Colin.	G035
22	Meeting on structure and information content.	G084
23	Exploration of interface design issues.	G007
23	Information structure first plan for City of Glasgow section - sections and sub-sections outlined.	G090
24	Analysis of structure of University information.	G096
24	Problems with sound, hard disk space.	G084
27	Analysis of information structure for the computer Science Dept, - rationale for hierarchical structure.	G094, G095
27	Early main menu page. outlined.	G093
27	INTERFACE report completed. Some maps scanned.	G084
28	Early report to INTERFACE (client).	G043
28	Report to INTERFACE - contains some structuring options, interface outline.	G091
March		
2	Decisions about colour, display sizes.	G084
3	Further concrete design decisions - no details.	G084
10	Structure and interface virtually complete.	G084
10-16	Work on presentation to INTERFACE.	G084
16	Graphics problems and decisions, sound problems - no suitable sound editing software available.	G042
23	Further log opened covering latter part of the first phase of the project.	G085

Date	*Event*	*Evidence:*
1994		
March 24	Video filming - problems with lack of activity in labs and outside, and large amounts of the University being scaffolded.	G084
25	Reviewed footage	G041
April		
13	Stuart and Donald took photographs of various landmarks in and around the University campus.	G041
16	'Filled in structure with lots of buttons and backdrops. Some taken from network or hard disk, some designed by us.'	G041
19	Querying Mac to PC conversion. Stuart and Donald took photos for digitisation. Letters sent to Scottish education departments to find out more about the facilities schools have. Photo session organised to get some staff photos. Ongoing design decisions re interface. Main structure complete ready for addition of video.	G085
20	'Stuart, Donald and Colin redesigned title page (again). Colin tidied up framework of system, making it more modular; also labelled each module and a few backdrops, buttons and links.'	G041
28	Arranged a date with Mark Sanderson for testing prototype on the CD-ROM simulator.	G085
30	Stuart scanned more pictures in for use in the prototype and extended the prototype.	G041
May		
3	Disk housekeeping.	G085
17	Paul scanned in some photos and information leaflets on transportation and put them into the interface.	
19	Using Adobe PhotoShop, Paul drew up some diagrams for the transport section and entered them into the right section.	
20	Paul found a book on Glasgow history and some photos were scanned. A couple of pages were typed in to finish of this section.	G041

Date	Event	Evidence:
1994		
June	Much documentation this month, probably for assessment purposes. This is not considered as integral to the project.	
4	Meeting to discuss documentation and remaining tasks.	
5 - 12	Worked on tasks identified. These included putting in sound, grabbing video footage, digital camera input. The system was by now already set up ready for the inclusion of these elements. Testing and ensuring the prototype is consistent was seen as the main work here. Also, running a CD simulation. Derek and Colin allocated task of doing the documentation. Incorporated CD photo images successfully to give good quality backgrounds.	G041
6	Authoring of extra text content begins for second phase - full version of prospectus.	G048, G052
7	Started digitising video using Fusion Recorder - 'finally got Fusion Recorder going'. Paul experimented with frame rates and numbers of colours. Noted the huge amount of memory consumed by more than a minute of QuickTime. Donald tidied up the opening sequence.	G042
8	More video digitisation, from videos supplied by Media Services. Problem definition rewritten by Colin as the existing one now looked very dated.	G042
9	More information collected from education departments about educational computing facilities in Strathclyde region. Started adding sound to the video sequences.	G042
12-14	Demonstrations of prototype to teaching staff	G041
July		
7	Initial design for interface of 'Guided Tours' section	G122
7	Problems in linking, link representation, particularly remembering the outcome of earlier discussions about this, and Director scripting	G104
8	Further work on structure	G103
13	Decisions on some colour elements	G072
23	Consideration of tasks uncompleted, problems to solve.	G069

Date	*Event*	*Evidence:*
1994		
July 23	Tasks still to do at this point include addition of Ordinary Degree pages, interface design elements such as page transitions, cursor changes for certain screen areas, buttons to make, video to add, some pictures to import for 'Guided Tours' section.	G105
August		
16	Text authoring - menu items	G135
23	Text authoring - accommodation	G106, G 107
25	Text authoring - Glasgow amenities	G110
29	Text authoring - sports facilities	G111, G 112
31	Text authoring - entertainments & amenities	G108
September		
8	Text authoring - projects	G137, G138
8	Text authoring - sample projects	G055
8-14	Text authoring - courses	G141-153
8-14	Text authoring - individualised examples	G114-G121, G124-G130
13	Text authoring - course objectives for Computer Science	G140
14	Text authoring - entertainment	G050
14	Text authoring - Glasgow transport	G049
14	Text authoring - history & location	G109
14	Text format rules adopted	G070
November		
22	Report on debugging and checking for inconsistencies	G113
8	Text content - class exam	G133
4	Text content - courses	G134
8	Text content - City of Glasgow	G132

WHAT THE CHRONOLOGY TELLS US

The chronology gives a good picture of the overall course of design and development. In the first phase, it shows an initial concentration on aims and resources followed by a more varied set of tasks for building the first prototype, including structural design interface design, and graphics issues. The final part of design concerns a range of problems, and the addition of video elements. The second phase begins with the resolution of some structural and interface design problems, followed by an extensive programme of text creation and import.

WHAT THE CHRONOLOGY DOES NOT TELL US

A number of tasks are more or less invisible. User analysis is mentioned, but not detailed. Task analysis is not apparent, although the 'scenarios' mentioned are partly an attempt at this. The information design process is not elucidated. Also invisible is the redesign of the interface that clearly took place between the first and second versions of the VOD. These faults are largely a product of the evidence available.

Appendix 3e: GT06 - interpretation and answers to Levei 2 questions

This table shows the interpretation of material in Table GT05 and the chronology, as answers to Level 2 questions. References are to the individual Level 1 questions and to the computer files, the latter being numerals prefixed with a 'G'.

1. What design guidance and methods were applied by the developers of the system (if any)?

The design team did not use a formal design methodology (GM1). Certain techniques adopted from software engineering and systems analysis were adapted for the project, including a formal problem definition (GM15), requirements analysis (G005), feasibility study (GM6), structure diagrams (G016), an analysis of users (G004), and contextual analysis (G003). Used examination of other products to provide guidance as to what to do and not to do (G002, G045). Not all of these were formally completed at the time they were required - many were completed retrospectively.

There is some evidence of the use of user interface design principles (G012, G005), although the interviewee suggested that 'common sense' was the main guide (DD8). The principles were as follows:

- Design for ease and pleasure of use
- Design for consistency
- Design for a non-modal system
- Design for structural visibility
- Design for flexibility
- Design for direct entry to information
- Design for extensibility
- Design for reversibility of actions

The source is unknown.

In general, the design approaches taken appear to have been influenced by the team's earlier training in computer science and software design.

2. Which of these were useful, and to what degree?

It seems that the analysis techniques had a complex role: 'it wasn't linear, it was, you had to work backward and forward through it,' (GM6) - so some of the analysis was written up retrospectively, which suggests that such techniques as requirements analysis and specification were limited in their usefulness. The interviewee did not voice any misgivings other than this. This is of course broadly in keeping with other evidence of software - and other - designers at work. No structured analysis was used. The team concluded that:

"Although the structure of the system we are developing will be complex and of central importance to the overall end product; structured analysis is limited because there is no data flowing through the system. Each screen full of information is displayed when the link to it is activated. There is no data processing taking place at all; other than referencing different pages. This makes the use of Data Flow Diagrams and other forms of structured analysis unnecessary." (G016)

The user interface design principles were somewhat general, and of a kind that is open to much interpretation, and it is therefore difficult to say for certain that they were followed to any great extent. The interviewee's reference to 'common sense' (DD8) suggests that perhaps they were not taken too seriously. Such principals as 'design for a non-modal system', and 'design for consistency appear to have been followed (G012, GCD01). It is clear that they are also aware of the problem of 'cognitive overload' despite its non-appearance in their list of design principles (G012). This suggests the integration of a wider set of principles and guidelines on an informal basis. This is perhaps the basis of 'common sense'.

3. What was the design process?

The project started from an idea by Richard Cooper, a member of the Computing Science Department's academic staff (supervisor) (GC1). The clients were Phil Gray, also a member of the Computing Science staff, Catherine Lyons, their information officer, and a group within the department responsible for marketing and student recruitment, the INTERFACE group (G091, GM6). An initial problem definition was worked out (G098). This shows a four-stage development was anticipated, starting with a review of the existing electronic prospectus, followed by an early design, a prototype and user testing, followed by a full implementation using leading-edge technology. This was followed by a requirements analysis defined on the basis of analysis of existing software (GC10, G02), contextual analysis (G003), and input from Catherine Lyons and Phil Gray. A somewhat limited user analysis was conducted, which was based on personal ideas as ex-applicants, and contacts with schools and education departments in Scotland (G04). No formal task analysis appears to have been conducted. An analysis of the information content was made by examining an earlier version of an electronic prospectus for Glasgow University made in HyperCard . This appeared unsatisfactory, and the information content and structure was rethought. They assessed the relative balance of the various elements in the existing HyperCard stack , and weighed this against their existing ideas about the material (GC12). Then they integrated this with the idea of a Virtual Open Day that Phil Gray came up with, that is, they adapted the information content of the existing stack to fit in with this structure. A requirements analysis was produced. D005 shows a requirements analysis in terms of 'who, what, where, when, how' basis. This is followed by a formal requirements statement listing tasks the system is to perform; a system specification in terms of features, including some general principles and guidelines; functional and non-functional requirements, hardware characteristics, error handling, quality issues etc. There is no evidence as to how the design team was chosen, and one assumes that as it was a student project they were already friends and acquaintances. Information content and resources were based on the information analysis earlier. Choice of hardware was based on hardware available in Scottish schools - predominantly Apple Macintoshes (DR3, G031). A decision was made to use proprietary authoring software. This was based partly on personal interest - they were 'sick of hacking out code', and becoming interested in information systems as opposed to software engineering. It was also simpler to use the ready-made features of Director than trying to write routines for everything from scratch (DD 12, DD14). Software choice was initially constrained by the choice of the Macintosh platform, and narrowed down from a list of four products, HyperCard, SuperCard, Authorware Professional and MacroMind Director, to MacroMind Director, largely on the grounds of superior video sound and colour handling (DR4, G029). CD-ROM was chosen as the distribution medium on the grounds of the likely size of a document containing digital video, costs of production and reliability (DR5, G029). Director version 3 was used for the prototype, and version 4 for the second phase, the main VOD.

A number of other items of software were used during the project (DR7). These included PhotoShop for editing and manipulation of still images (G028) Simple Player, MovieShop, Fusion Recorder (G030) and Premiere (G027) for video capture, review and editing, Sample Edit for sound editing, Microsoft Word, Claris Works, PhotoCD (G026). A constraint was that there was only one copy of Director available. This meant that it was important to have alternative software for reviewing and editing video, hence the number of different video capture and play software items used (DR7).

A limited version of prototyping was used (DR7, G032). There was an initial prototype, which the team considered as part of a feasibility study into the possibility of a full electronic prospectus. Much work done on paper due to lack of availability of Director at first (DD3). This prototype was completed to the extent that all departmental information was included, but little general information. No extensive evaluation took place. The prototype was then largely rebuilt using the same basic concepts and many of the interface features (G086). In July 1994, work commenced on a full VOD prospectus, a more polished version with much more extensive material, including sections on the University and the city of Glasgow. At this stage, the focus was on the general feasibility of the project, especially in terms of technical possibilities and the kind of presentation of information that was possible (DD1)

Design resources were limited to design advice from Phil Gray (DR8), and the assistance of a photographer from the Media Department. There was no assistance with graphics or information design.

The information content was described as relatively unimportant in the prototype stage, as opposed to structure, appearance, navigation and control (DD1, DD3). However, the documentation reveals that it was considered as important in evolving an appropriate structure

(G088). The two cannot easily be separated. In the second phase, information content was either authored by the design team or by Catherine Lyons. (DD1). General design ideas came from experience of other software (DD2), and from the team's exploration of the authoring software and manuals: "And someone's been reading the Director book: 'Oh we can do this as well'." (DD3).

The system was designed from scratch, and did not borrow from either the existing paper prospectus or the original HyperCard prospectus (GC10). The overall design approach was a top-down one (DD3). User interface design was considered at a fairly early stage (February - G007). A series of modules were designed using a shallow hierarchical structure of main menu and sub-menus (G018, DD5), and the information content and other material filled in. The exception is a rather complex cross-linked section dealing with course options (GCD01, G043, G092). Shown, and this seems to confirm the way the way they were designed, as modularised units and sub-units in G092. This series of diagrams was the main design technique. The rationale behind this choice of structure was to simplify navigation, which the team had seen as a problem with the original HyperCard stack (G002,). Apart from this, the indications are that design was a fairly flexible and loose process. There appears to have been no design documentation for the second phase, the construction of the main VOD (DD18).

The second phase proceeded by section. The document was structured into several sections made from separate Director 'movies' linked together. Skeleton structures were made for each module, then filled them in (C2).The Department section was authored first, because it was most difficult (C1, G048, G052), followed by the City and University sections, which were worked on concurrently with the scripting for the Department section (C1, G106-G108, G110-G 112, G135).

The VOD prospectus was evaluated whilst the second phase version was in a part-finished state (DD15). Evaluation took place via a questionnaire and protocols conducted using prospective students at the Departmental Open Day, with the aid of psychology students (DD15, DD16, G013-015 - results, questionnaire and protocol tasks). The main priority was ease of use (DD16). Tasks and questions focused on navigation issues, picture quality, movie quality , effectiveness and value as conveyors of information, use of sound in gaining users' attention, general graphical appearance of the system, general effectiveness of the system, and suggestions for improvements and general comments on the system. This revealed a number of problems, apparently mostly relating to movie handling, although the related document is incomplete (G015). There was also some informal evaluation via schoolchildren brought in by Catherine Lyons (GC9).

Documentation was extensive but somewhat unstructured (G001-153). Some elements were well-documented, for instance there was a fairly detailed log for the first phase of the project (G042). There was a report to the INTERFACE group at an early stage, a project report at the end of the prototype phase, and after that documentation is limited to working notes such as notes on colour codes (G072), formatting decisions (G070), problems and un-implemented ideas (G069), and bugs and missing items (G113). No master documents were maintained for files, links etc..

Debugging was partly based on the testing at Open Day, and allowing stakeholders and other members of staff etc. to play with the system. Also the separation of the team into different sections enabled a fresh view of each other's sections that showed up faults (C6a). Lists were made of some of the problems and un-implemented items, eg. G069, G104 and G113. Some evidence of link checking, for instance in G104, but there is no evidence of a systematic approach to this.

4. How were design guidance and methods applied in practice?

Storyboards and structure diagrams were evolved before the Director software was available (DD3). Several structural options were worked out in diagrammatic form (G091).

User interface design principles were incorporated into the requirements document (G005). It is not clear how they were utilised in practice, but the interviewee's reference to the use of 'common-sense' rather than any specific principles suggests they were not used rigorously (DD8).

The requirements (C005) document does not appear to have been used as effectively as it might have been. It is comprehensive and contains some quite specific requirements, which could have been used as the basis of evaluation, but the evaluation tasks and questionnaire do not directly correlate with the requirements.

5. What design problems were experienced, and how were they resolved?

A number of problems were encountered. To begin with, the software was not available, and it was difficult for the team to know what was possible: "So it's kind of difficult, because you've got this sort of like an information gap between knowing what you want and not being sure what Director will actually do." (DD3).

Navigation was an early concern, navigational problems being apparent in the old Glasgow electronic prospectus (G002). The hierarchical structure meant it was possible to lose track of location in relation to the rest of the system (DD10). The response was to add a locational device in the form of a path-name at the top of each screen, and to constrain the options as much as possible.

A number of small-scale technical problems were encountered. These include problems with page transitions, keeping the Director score and cast tidy, and setting up the design so it could easily be changed (G069). Also problematic was the layering of objects. This did not appear to be amenable to a structured approach (C1). The structuring of the document into different movies for each section caused problems with sound continuity.

Both the hardware environment, in terms of screen size, and Director imposed certain constraints in terms of the amount and size of text and graphics that could be shown on screen. Problems in chunking material were not directly referred to although this might be an expected consequence of such constraints.

It had originally been intended to include a full search engine; but this was also found to be technically too difficult to be accomplished in the available time, so the search facilities in Director were utilised. This made it necessary to anticipate in advance the kinds of things users might wish to look for. (DD12).

The most difficult parts were the Guided Tour section, a section which utilised a lot of video and had quite a complex structure which caused a lot of problems with the users' routes (C1, G104), and with scripting. There is a trade-off between the desire to keep the script as tight and compact as possible, and therefore faster to run and less bug prone, and ease of implementation (C104):

"...I am thinking mainly of the course map to general pages - this will involve disabling all the sprites that represent the guides; supplying some default context and removing the signpost icon. It seems much easier just to duplicate the info (and the layout?) elsewhere." (G104).

Video was identified as a significant problem area, as already noted. Many video problems related to how to give users control. This led to a number of arguments with regard to how the user should be able to control the video sequences, in particular making sure that videos reset to the beginning. The options were to use the standard Apple video controller or a tailored version. In the end a tailored version was chosen (C3). Another aspect was the problem of getting video compressions right - getting an acceptable combination of utilisation of disk space, smoothness of movement and detail of image, and getting this right for the lowest specification of machine - in this case the Macintosh LCII. This entailed running all video on an LC2 to ensure success.

Sound was relatively limited in the VOD prospectus, one reason for this being the lack of a sampling utility which could deal with longer sequences (G034). Another was the failure to gain the co-operation of the Department's sound specialist (G034). It was otherwise fairly straightforward except where the VOD was separated into different 'movies', although its potential to annoy was noted and a 'music on-off' button included for the developers as much as the users (C3).

Other problems were organising the team and allocating work, and version control - keeping everyone working on the last or best version of the project (C6). Another point is the learning curve associated with the software packages concerned. Where multimedia elements are concerned, the designer may be called upon to use a number of fairly substantial and complex software packages, and this was evident in this project, where the team had to use Director and Premiere. The reluctance to engage with another complex piece of software meant that in the prototype stage video was simply cut and pasted in from the video capture utilities without any editing (DR7).

In summary, it seems that there was limited outside expertise directly involved (GC4). This created a context in which problems had to be resolved by the team or features could not be implemented, as in the case of sound. Within this context, the main problems appeared to be technical although it is possible that this may be a distorted perspective, a consequence of the team's scientific/technical orientation as computing science students. These problems were frequently caused by the limitations imposed by the Director software. Choice of software is clearly critical. Although at the time the VOD was made, Director was probably as good a choice as any for this kind of project, it obviously posed many awkward problems. Given the present state of authoring software, it is probably a question of choosing which problems the author wishes to deal with: there are no perfect solutions. It appears that design is most severely constrained in the first instance by not knowing what is genuinely achievable with the authoring software available, and then by the various technical obstacles imposed by the hardware and software. In many cases where problems arose in this project, it seems that they were resolved by constraining ambitions to fit in with the hardware and software.

Appendix 4

General introduction to the Hypertext Authoring and Design Methodology

Introduction

This methodology is intended to assist the author of small to medium sized hypertext and hypermedia documents to design such documents as effectively and efficiently as possible. 'Small to medium sized' is difficult to define. In digital terms, it depends greatly on the media forms incorporated in the document, and this itself is changing as software, hardware and compression techniques improve. In this context it is taken to mean hypertext documents approximating to anything up to a book length of approximately 200-350 pages.

The design methodology introduced here is less than prescriptive: it does not, and arguably should not and cannot, tell the hypertext author exactly how to author a hypertext document. What it does do is provide a way of breaking design into manageable stages, and organising those stages. Within each stage it provides documents that support design and development, in the form of general guidance, task lists, and documentation aids.

The philosophy behind this approach to design is based on the idea that hypertext and hypermedia design is best done on the basis of a good understanding of both the user and the task(s) for which they will be using the document, and of the knowledge to be contained within the document. Such understanding can best be gained by means of an iterative approach to analysis of the users, task and the resulting requirements. Hence although tasks are organised according to stages, these stages may not necessarily be completely finished in sequential order, but may be revisited later in the design process in order to incorporate new information about the users, their tasks and the knowledge domain contained within the hypertext document. The work on which the design methodology was based took place in the context of information provision, and the methodology is oriented towards this kind of function, but should be applicable to hypertext for a range of other tasks as well.

The contents of the methodology and their relationship are outlined overleaf.

Sources

The relationship of this methodology to earlier work has been discussed in Chapter 7. The methodology is a synthesis of work by the author and others. Where individual principles and guidelines based on the work of others is cited directly, the source is acknowledged. In some cases, material from the other sets of guidance given in Appendix 1 has been combined and modified. Acknowledgements are not given in these cases, but are due to the authors whose work has been re-presented in Appendix 1.

The methodology elements[1]

The Hypertext Authoring Design Methodology has three main elements: procedural overviews, task lists and a set of forms for records and reports. First there are procedural overviews. There are two of these, one showing the methodology as a sequence of stages. This gives a possible sequence for design. However, given the need for continued evaluation, and the close relationship between

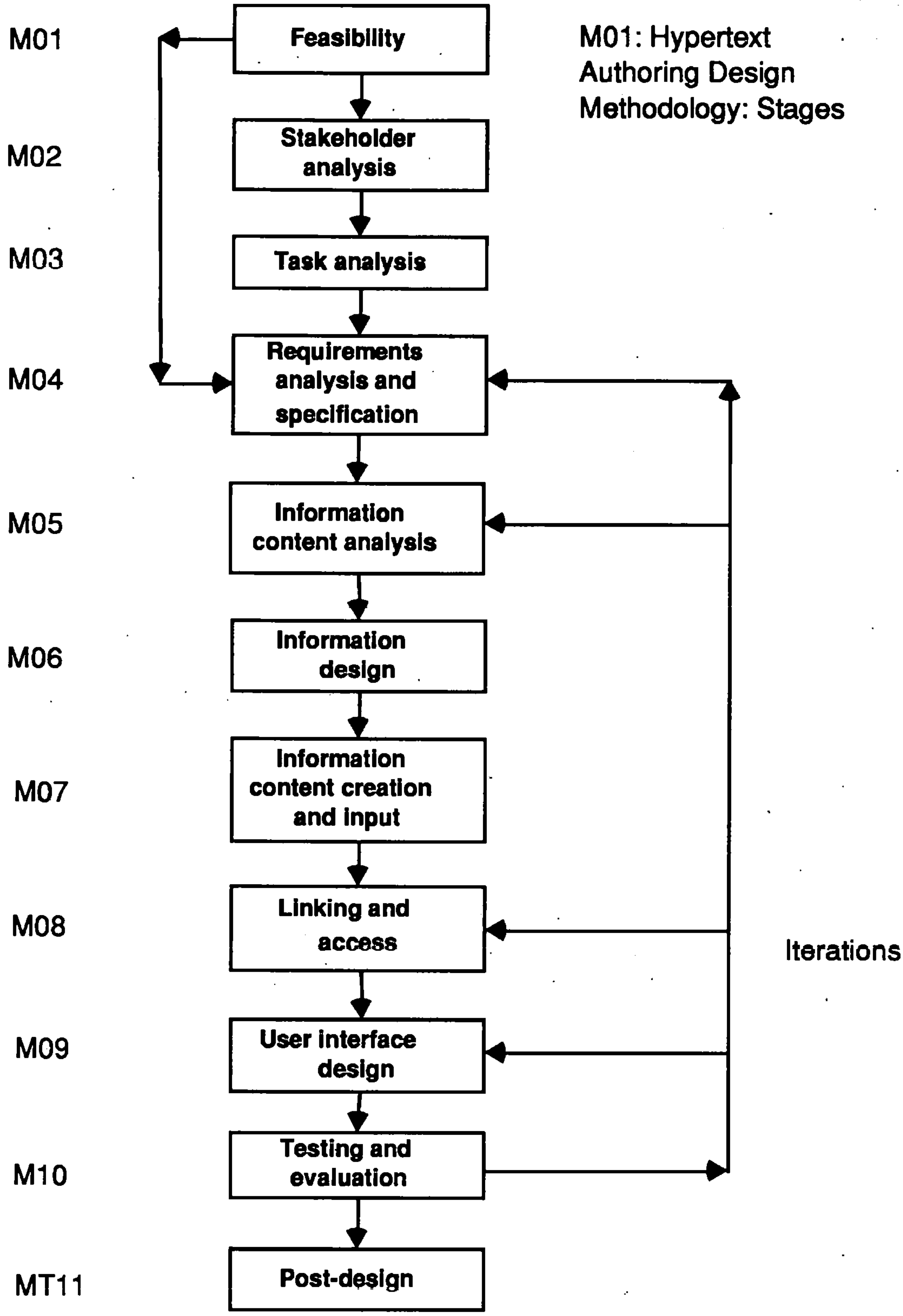
---

[1] A published version would be bound loose-leaf for copying of forms, and would also include word-processor templates for forms.

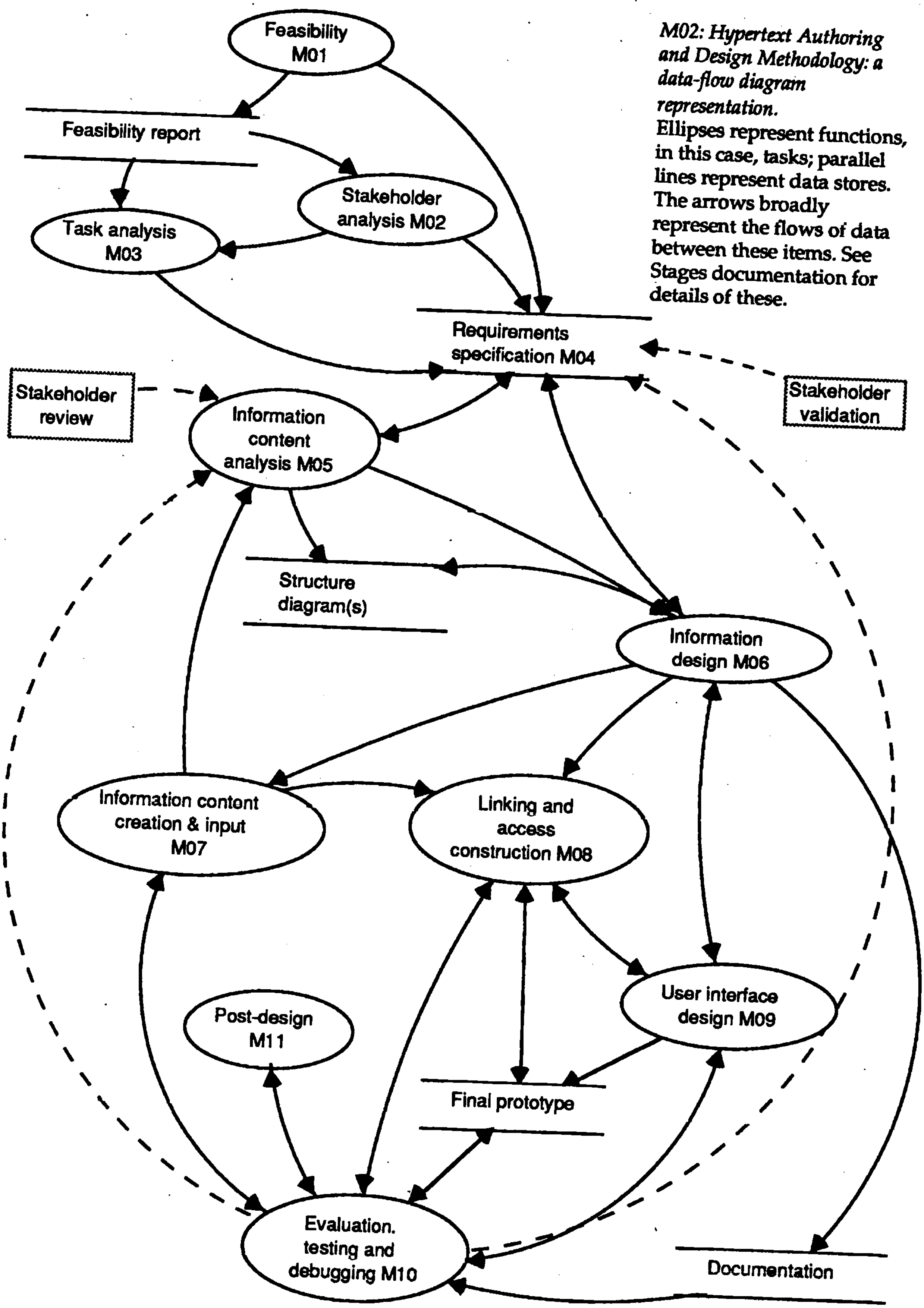
such tasks as information content analysis and information design, an alternative view of the design procedure is given. This is in the form of a data flow diagram, a technique used in areas such as database design.

Secondly there are task stages, MT1 to MT 10. These give details of the tasks to be performed at each stage, plus information on inputs and outputs at each stage in the form of data and prototypes.

The third element is a set of forms included to encourage a methodical and properly recorded process through the design work. This should enable the proper completion of tasks and the keeping of a complete record of the project's progress, the document structure and interface, and of those involved and their tasks.



M01: Hypertext Authoring Design Methodology: Stages



M02: Hypertext Authoring and Design Methodology: a data-flow diagram representation.
Ellipses represent functions, in this case, tasks; parallel lines represent data stores. The arrows broadly represent the flows of data between these items. See Stages documentation for details of these.

Stage MT1: Feasibility

Tasks

Feasibility can cover a wide range of items. The extent of this range will be determined largely by the centrality of the document under consideration in the organisational context: the more important the document the more extensive the feasibility report. The main concerns of the feasibility report are the technical, economic and operational feasibility of the document.

1. Determine the overall objectives of the document and put them in writing.
2. Consider what information will be covered by the document.
3. Market analysis: identify market niche & potential users. List user problems, needs and issues which should be addressed.
4. Perform a cost-benefit analysis
5. Agree a production schedule and costs
6. Investigate information sources, availability and permissions
7. Identify media and distribution channels to be used
8. Determine who will create the document: identify potential authors and designers with relevant skills. These may include:

- Copy editor
- Graphics artist
- Sound/video technician
- Hypertext co-ordinator/designer
- Reviewers
- Subject matter experts

9. Identify hardware and software constraints
10. Identify resources :

Software resources

Consider need for and availability of:

- Hypertext authoring software
- Graphics software
- Animation software
- Spreadsheets
- Word-processing software
- Spelling checker
- Grammar checker
- Electronic publishing software
- CD-ROM mastering software
- Video capture and editing software
- Sound capture and editing software
- Programmable software for intelligent documents

Information resources

- Central glossary

- Central collection of acronyms
- Central collection of diagrams
- Document templates

Outputs from this stage:

MT1a: Statement of objectives; MT1b: Resource analysis

Stage MT2: Stakeholder analysis

Inputs from previous stage:

MT1a: Feasibility report; MT1b: Resource analysis

Tasks

The object of this stage is to identify the various stakeholders and their requirements and interests. Stakeholders are *all* those with an interest in the proposed document, not just the end users and client. Most significant here are the users and client, but others affected by the development, use, maintenance and updating of the document, such as administrative staff should also be identified, as they may have a key role in the success or failure of the proposed hypertext document.

1. Identify and list all stakeholders. The feasibility report (MT1a) already identifies the client and general user group.
2. Identify users and their tasks: who performs which tasks, why and where. Anticipate the broadest user profile.
3. Identify the interests of other stakeholders. Some of these may already be identified on MT1b, the resource analysis.

Techniques:

Scenarios: from the list generated, develop scenarios for the possible use of the hypertext document. Why will they be reading the document? What do they already know? What do they need? How old are they? What are their physical characteristics? If possible, show the appropriate scenarios to a few prospective users and other stakeholders.

Outputs from this stage:

MT2a: Stakeholder analysis; MT2b: User analysis.

Stage MT3: Task analysis

Inputs from previous stage:

MT1a: Feasibility report; MT2a: Stakeholder analysis; MT2b: User analysis.

Tasks

Task analysis is the process of identifying and recording the tasks performed by users of systems. It is a huge subject, and a number of systematic task analysis methods exist. These have particular strengths and weaknesses, and no particular analytical method is advocated here. Some general principles are identified, and a selection of techniques for acquiring task data are listed.

1. Clarify the purpose of the analysis on the basis of general objectives and results of scenarios.
2. Perform task analysis - select suitable task analysis techniques:
   - Group discussions
   - Interviews with users
   - Interviews with subject area experts
   - Task observation
   - Walk-throughs
   - Verbal protocols
   - Pattern noting
3. Represent the analysis. The appropriate form for this depends on the type of task. A wide range of possible techniques are described in Kirwan, B. and Ainsworth, L.K. (1992): A guide to Task Analysis (London: Taylor & Francis)

   Some principles to apply (from Johnson, P (1992): Human-Computer Interaction: Psychology, Task Analysis and Software Engineering. London: McGraw-Hill.):
   - Check the analysis with the task performers
   - Analyse more than one person
   - Make use of more than one technique for gathering knowledge

   More information on task analysis can be found in 'A guide to Task Analysis', by B. Kirwan and L.K. Ainsworth (Taylor and Francis, London 1992).

Outputs from this stage:

MT3a: Task analysis report

Stage MT4: Requirements analysis and specification

Inputs from previous stage:

MT2a: Stakeholder analysis; MT2b: User analysis; MT3a Task analysis report

Inputs from iteration

MT5a: Unprioritised information; MT5b: Prioritised topic list; MT5c: Finalised topics list.; MT6a: Structured topic list; MT6b: Skeleton structure diagram; MT6c storyboards.

Tasks

Requirements analysis is often an on-going process. The analysis done so far is an important component, but the designer can expect to have to modify requirements on the basis of increased knowledge of the users and the design situation generally.

The accompanying form, MT4a, suggests some of the areas which will need to be included. These should be tailored according to the objectives of the document and to the nature of the users and their tasks.

On the basis of previous analysis, determine :

Functional requirements:

- The information requirements
- Navigation and access requirements

Non-functional requirements:

- Usability requirements

Outputs from this stage:

MT4a: Requirements specification

Stage MT5: Information content analysis

Input from previous stage:

MT4a: Requirements Specification

The Information Content Analysis stage covers a range of tasks, from gathering the information identified as needed at the Requirements stage to structuring that information.

1. Gather information consulting users, subject matter experts and other sources as required. Much of this material may well already be available via the Requirements Specification MT4a.

2. Analyse information sources:

a) identify list of topics and components from documents and user requirements (MT4a). Use form MT5a. Do not structure this material, but aim to capture all elements.

b) condense and prioritise this information. Be guided by the requirements specification as to priorities. Use Form MT5b.

c) establish the initial structure of the information: identify sequences, hierarchies and other relationships. If these are not yet apparent, conduct further task analysis or consult users and subject area expert as required. Use Form MT5c.

3. Add outline contents and prioritised form of the material to Form MT5c

4. Get reviews from the various stakeholders

Outputs from this stage:

MT5a: Unprioritised information; MT5b: Prioritised topic list; MT5c: Finalised topics list.

Stage MT6: Information design

Inputs from previous stage:

MT4a: Requirements specification; MT5a: Unprioritised information; MT5b: Prioritised topic list; MT5c: Finalised topics list

Inputs from iteration:

Final prototype; MT9a: Log of interface design decisions

Tasks

This stage involves designing the document. Information design for hypertext functions at a number of levels. At the structural level, design concerns eliciting the logical and semantic relations between the various topics and representing these as a linked structure. At a lower level, the author is concerned with designing individual screens to present the information, a process that is inseparable in a practical sense from user interface design.

The document should be designed around the structuring and presentation of the information, and this should reflect the information requirements of the users. Avoid committing to a structure prematurely, and do not allow the idiosyncrasies of the authoring software to dictate structure.

1. Decide whether to convert existing material or not. Remember that this is never as easy or as cheap as it sounds.
2. Create the initial document structure. Design top-down in the first instance. Establish a skeleton structure based on the information's structure elicited in MT5 (MT5c) and validate with users and subject area experts. Also check with the earlier information on Forms MT5a and MT5b. Although a form, MT6b, is provided for the skeleton structure diagram, it is likely that this will be too large to fit this except in overview form. It is useful to use a suitable graphics package here as the process of structuring is iterative. Keep checking back with users whilst evolving the structure.
3. Fill in detailed structural relationships including any cross referencing to related concepts. Again, check with users.
4. Make decisions about size of information chunks for each topic node. If information is likely to be printed out, size chunks to fit standard paper page sizes. Consider the user interface design at this point: decisions about display size, screen layout and typefaces all affect the amount of information that may be displayed for each node. Use storyboards to develop layout (MT6 storyboard template).
5. Produce a skeleton document on the basis of the structure diagram.

Outputs from this stage:

MT6a: Structured topic list; MT6b: Skeleton structure diagram; MT6c storyboards.

Stage MT7: Information content input

Inputs from previous stage:

MT6b: Skeleton structure diagram; MT5a Topic list; MT5b: Prioritised information; MT5c: Structured topic list; MT6b: Structure diagram(s); MT6c: Story boards.

Inputs from iteration:

MT10a: Test Reports.

Tasks

Some material may already be available from earlier, and this will require scanning, editing and file conversion as appropriate. Some material will need to be written specially.

1. Determine what media are necessary
2. Liaise with media professionals eg film crew, graphic artists, animators.
3. Prepare material: Collate and check existing material. Collect or create additional material.
4. Arrange for editing of text and graphics. Send out drafts for amending if necessary.
5. Arrange for conversion to appropriate digital form
6. If the authoring software you are using allows it, develop and use a style sheet. If not, specify the formats, type faces etc. on the enclosed form (M7a).
7. Secure legal permissions
8. Finalise text & approve with subject matter experts, client and users
9. Fill in the nodes
10. Work from a master topic list using form MT5c: maintain a list of topics, including name of original author/source, original file name and format, date of completion.

Outputs from this stage:

Document prototype; MT7a: Information style sheet; MT5c (amended)

Stage MT8: Linking and access

Inputs from previous stage:

Document prototype; MT4a: Requirements specification; MT5c (amended); MT6b: MT6b: Structure diagram; Document prototype; MT7a

Inputs from iteration:

MT10a: Test Reports.

Tasks

1. On the basis of the requirements analysis (MT4a) and structure diagram (MT6b) identify the linking options which will be required to allow the user to accomplish all the tasks identified.
2. On the basis of the requirements analysis (MT4a) create support for user navigation of the document. Strategies here include footprints and overviews, but the use of simple devices such as different typography for different kinds of material, and numbering of sections and nodes should not be overlooked.
3. Construct links

   Links can be created in several ways. The options are to:

   - Manually construct links
   - Compile links from tagged files.
   - Computed links.

   The decision depends to a large extent on the time and labour available, the software used for authoring and the technical support available. Manual construction is an option with the majority of authoring approaches, but is laborious. Compiled or computed links may not be available as an option with some authoring packages. If using compiled or computed links, expect some editorial input in any case.
4. Record all links on form MT8a
5. Construct access mechanisms for direct retrieval
6. Add facilities for editing, annotation and printing as required
7. Prototype with users and modify accordingly

Outputs from this stage:

Linked prototype; MT8a: Record of links form

Stage MT9: User interface design

Inputs from previous stage:

Linked prototype; MT4a.

Inputs from iteration

MT10a: Test Reports.

Task

Although this stage is numerically late in the sequence, user interface design cannot be left until last. See Stage 7:

1. Dialogue design. This involves the following key areas:
   - Screen design
   - Node representation
   - Link representation
   - Formatting and layout
2. Confirm style guidelines as established on form M7a

   Use the guidelines accompanying this methodology as a basis for establishing your own user interface guidelines to fit your user analysis.
3. Confirm help required.
4. Identify the screen layout - workspace, control area, status area, message area etc. Use a standardised grid to maintain consistency.

Outputs from this stage:

Final prototype; MT9a: Log of interface design decisions

Stage MT 10: Testing and evaluation

Inputs from previous stage:

Final prototype (MT9); MT4a

Tasks

The approach to evaluation and testing adopted here is an iterative one, based on the recognition that requirements cannot easily be visualised by stakeholders and designers until at least some progress towards a visible artefact has been made. So although evaluation appears here at this late stage it is therefore an ongoing process which commences with validation of the requirements document with stakeholder , and continues through the design process.

1. Determine who will check, proof-read, debug, review and evaluate the document. This means identifying users and others: get as many people as is feasible within time and budgetary limits - users, stakeholders, subject matter experts - to use and review the document.
2. Review the proposed content and structure with potential users. Use an iterative approach - evaluate with users, change the document accordingly and re-evaluate.
3. Check the hardware and software
4. Spell-check and proof-read carefully
5. Re-check the links. Use the list of links to ensure a completed check (MT8a)
6. Debug. Get a fresh view. Bring in expert and non-expert reviewers and encourage them to find problems - with a reward if necessary.

   Test the help included and any printed instructions as well as the document itself. Check with users if the help system provided the kind of help required. If users appear to need help often, find out why, and change the document to prevent the problem.

   Validate the requirements document - then use it as a basis for testing systematically, both as a set of heuristics and with users.

   Use verbal protocols - get users to think aloud as they use the document.

   Use automated recording of usage if appropriate - but remember this can generate a huge amount of data, and reveals nothing of the users' experiences.

Outputs from this stage:

MT10a: Test Reports.

Stage MT11: Post-design

Inputs from previous stage:

Final prototype

Tasks

This stage groups together a number of assorted tasks that are often relatively neglected.

1. This really belongs to the previous stage, but - when you' think you've finished, check the document one last time. No matter how much you've checked before, check the production version again. Errors can creep in at the last minute.

2. Establish the procedure for ongoing updating.

Plan improvements: follow up users of the finished hypertext

3. Prepare acknowledgements and credits

4. Dissemination:

Develop package design and installation and use instructions: start documentation for use early and integrate with user interface design.

Provide consultation for problems

Outputs from this stage:

Completed hypertext document with instructions and packaging.

MT1a: Statement of objectives

Document:	Author:
Date:	Page: of

No.	Objective	Init.

MT1b: Resource Analysis

Document:	Author:
Date:	Page: of

Software

Hardware

Personnel

MT2a: Stakeholder analysis

Document:	Author:
Date:	Page: of

No.	Stakeholder	Init.

MT2b: User analysis

Document:	Author:
Date:	Page: of

No.	User	Init.

MT3a: Task analysis report

Document:	Author:
Date:	Page: of

User	Situation	Task	Goal	Init.

MT4a: Requirements specification

Document:	Author:
Date:	Page: of

Requirement	Init.
1 General goals:	
2 Overview	
2.1 Users: description based on MT2a	
2.1.1 Skill levels of users:	
2.2 Stakeholders	
2.43 Tasks: description based on form MT3	
2.3.1 System Specification	
2.3.2 Software requirements:	
2.3.3 Hardware requirements:	
3 Requirements Specification:	
3.1 Functionai Requirements	
3.1.1 Direct information access	
3.1.2 Browsing	
3.1.3 Links	
3.1.4 Graphics	
3.1.5 Video	
3,1,6 Sound	
3.1.7 Navigation	
3.1.8 Overviews	
3.1.9 Help facilities	

Requirement	Init.
3.1.10 Information content *3.1.11 Information structure* *3.1.12 Documentation* *3.1.13 Hardware considerations* *3.1.14 Performance characteristics* *3.1.15 Error handling* *3.1.16 Relationship to other systems* *3.1.17 Reliability* **3.2 Non Functional Requirements** The following are non-functional requirements for the document grouped under general headings. *3.2.1 User Interface and usability* *3.2.2 Aesthetics and quality*	

MT5a: Unstructured information	
Document:	Author:
Date:	Page: of

Information required (list all topics - unformated, uncondensed)

MT5b: Topic list

Document:	Author:
Date:	Page: of

Topic list (formatted and condensed data from MT5a)

MT5c: Structured topic list

Document:	Author:
Date:	Page: of

Structured topics (indicate headings, sub-headings, etc.)	Name of original author/sou rce	Original file name and format	Date of input	Date of completion

MT6b: Structure diagram

Document:	Author:
Date:	Page: of

Structure diagram

MT6c: Story board template

Document:	Author:
Date:	Screen number:

MT7a: Information style sheet

Document:	Author:
Date:	Page: of

Type Specification	Information type (headings, sub-headings...)	Init.

MT8a:Record of links

Document:	Author:
Date:	Page: of

Date	Link type	From (File)	To (File)	Init

MT9a: Log of interface design decisions

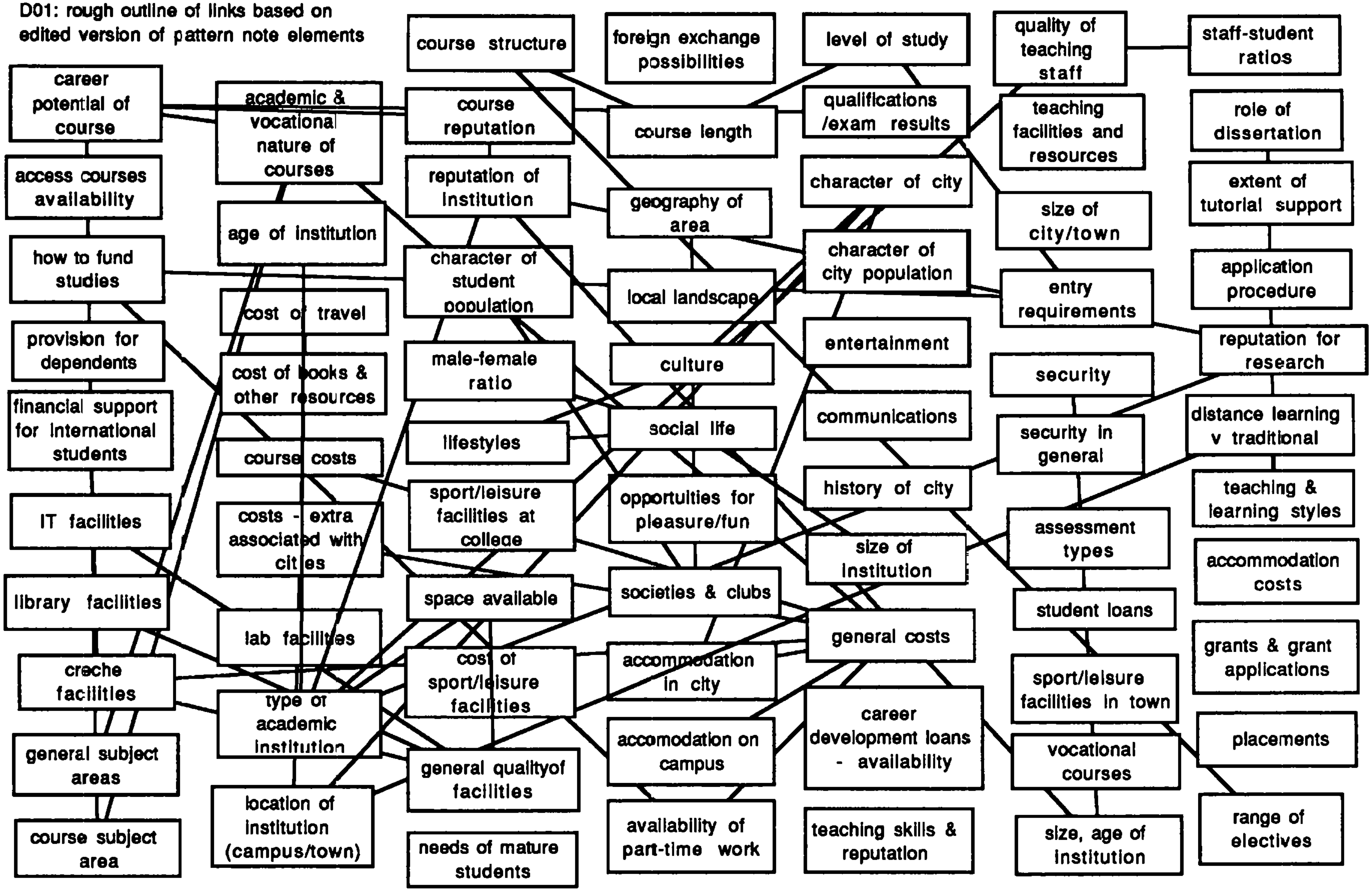
Document:	Author:
Date:	Page: of

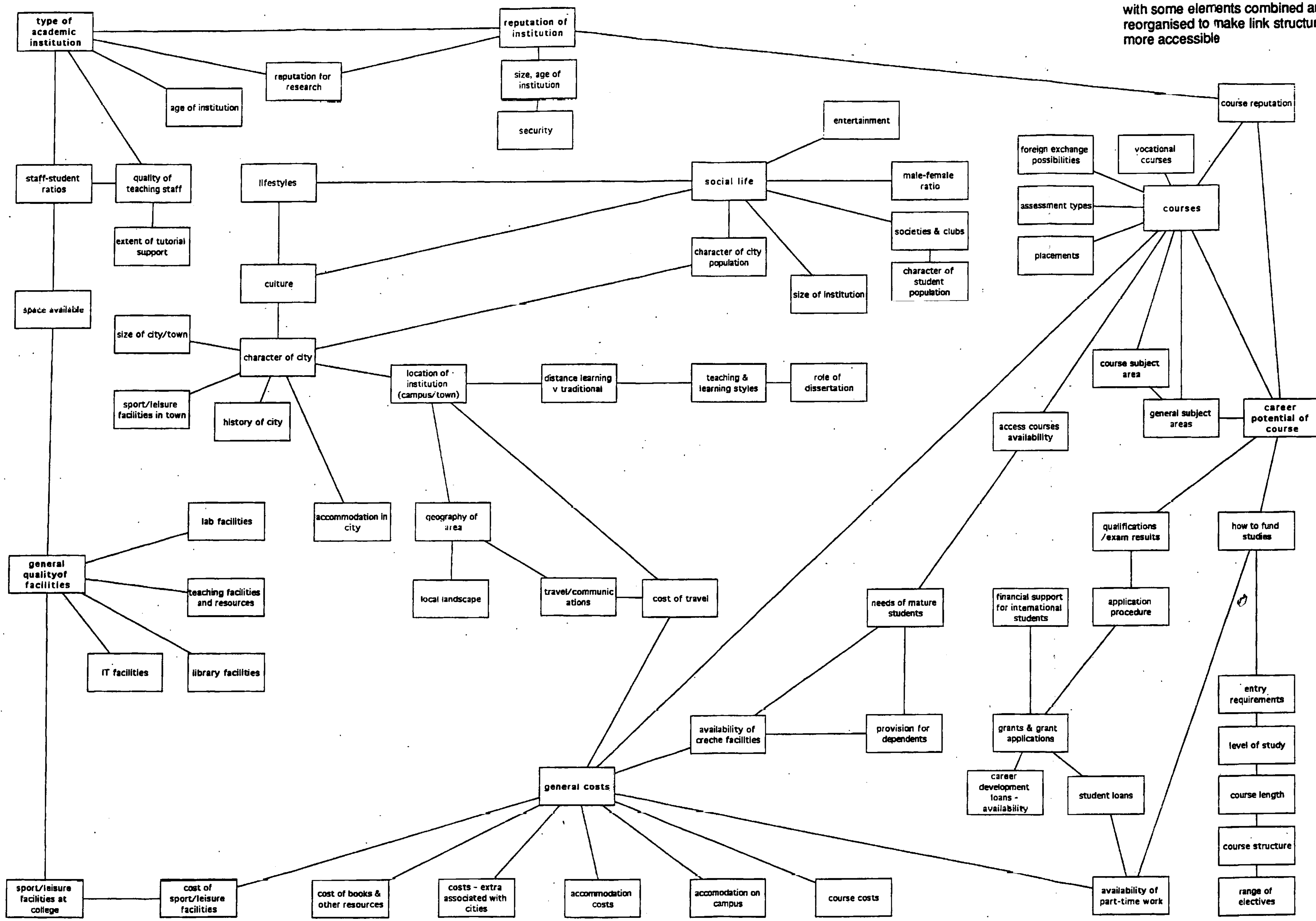
Date	Decision	Init.

MT10a: Test report

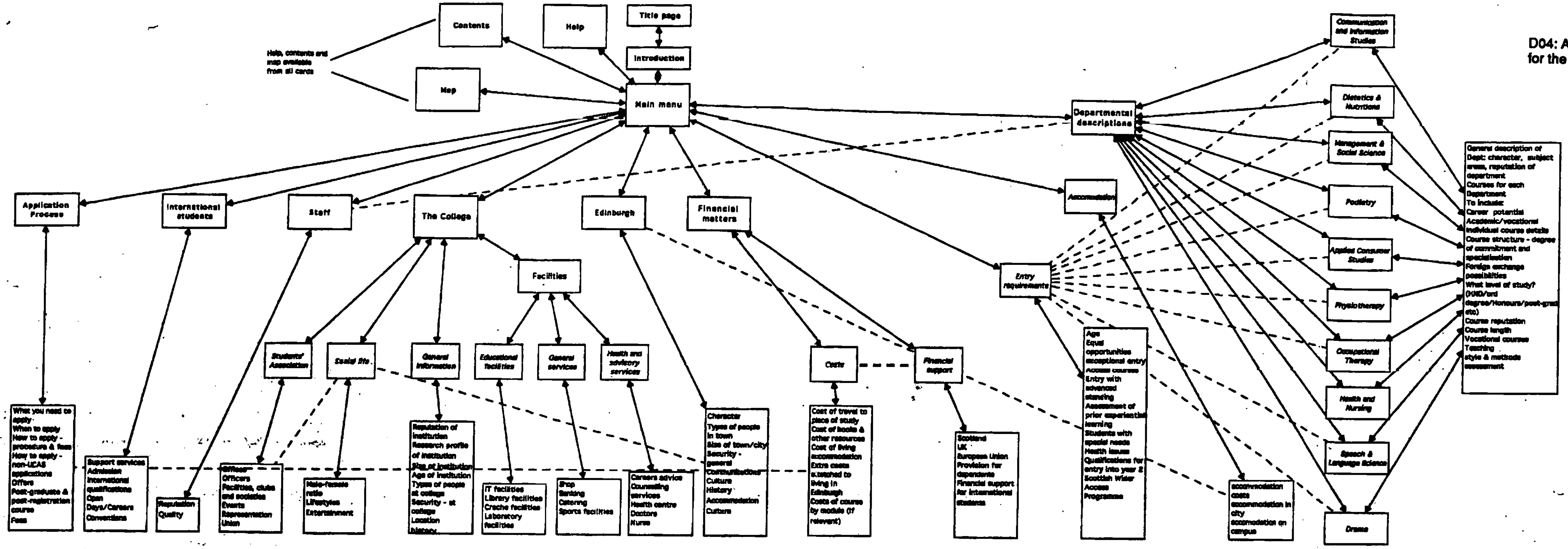
Document:	Author:
Date:	Page: of

D01: rough outline of links based on edited version of pattern note elements





D03: Structured links based on D01, with some elements combined and reorganised to make link structure more accessible



D04: A hierarchical structure for the QMC electronic prospectus